NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN ADAPTIVE APPROACH FOR PRECISE UNDERWATER VEHICLE CONTROL IN COMBINED ROBOT-DIVER OPERATIONS

by

Nicholas D. Valladarez

March 2015

Thesis Advisor: Noel Du Toit
Co-Advisor: Vladimir Dobrokhodov

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2015	Master's Thesis

4. TITLE AND SUBTITLE
AN ADAPTIVE APPROACH FOR PRECISE UNDERWATER VEHICLE CONTROL IN COMBINED ROBOT-DIVER OPERATIONS

5. FUNDING NUMBERS

6. AUTHOR(S) Nicholas D. Valladarez

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Joint robot-human operations potentially increase the efficiency, effectiveness and safety of the tasks they perform. The utilization of an autonomous underwater vehicle (AUV) as a robotic diver's assistant demands joint, dynamic operations involving precise physical interactions between an AUV, human divers, and the environment, which, in turn, requires a robust, accurate control system. A robot acting as a dive assistant would perform tasks such as tool carrying, worksite illumination, or other general assistance jobs that a "dive buddy" might perform. Such precise control of the AUV normally requires accurate knowledge of the vehicle's dynamics; however, this high level of accuracy is difficult to obtain without the employment of extensive system identification efforts. Additionally, the utility of the resulting model is greatly diminished if environmental conditions or vehicle configuration change frequently or unexpectedly. An ideal control system allows the AUV to switch between operational modes and objectives while accounting for uncertain environmental conditions, payload configurations, and possible failures of onboard actuators. Adaptive control has many applications in the underwater domain and can give a robotic diver's assistant the flexibility required to enable joint robot-diver operations. Therefore, two adaptive control system approaches, Model Reference Adaptive Control and $\mathcal{L}_1$ Adaptive Control, are investigated here for heave control of the Tethered, Hovering Autonomous Underwater System.

14. SUBJECT TERMS	15. NUMBER OF PAGES
unmanned underwater vehicle; autonomous underwater system; tethered; hovering; robust control; model referenced adaptive control; L1 adaptive control; parametric uncertainty; time-varying parameters	115
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AN ADAPTIVE APPROACH FOR PRECISE UNDERWATER VEHICLE CONTROL IN COMBINED ROBOT-DIVER OPERATIONS

Nicholas D. Valladarez
Lieutenant, United States Navy
B.S., United States Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2015

Author: Nicholas D. Valladarez

Approved by: Noel Du Toit
Thesis Advisor

Vladimir Dobrokhodov
Co-Advisor

Garth Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Joint robot-human operations potentially increase the efficiency, effectiveness and safety of the tasks they perform. The utilization of an autonomous underwater vehicle (AUV) as a robotic diver's assistant demands joint, dynamic operations involving precise physical interactions between an AUV, human divers, and the environment, which, in turn, requires a robust, accurate control system. A robot acting as a dive assistant would perform tasks such as tool carrying, worksite illumination, or other general assistance jobs that a "dive buddy" might perform. Such precise control of the AUV normally requires accurate knowledge of the vehicle's dynamics; however, this high level of accuracy is difficult to obtain without the employment of extensive system identification efforts. Additionally, the utility of the resulting model is greatly diminished if environmental conditions or vehicle configuration change frequently or unexpectedly. An ideal control system allows the AUV to switch between operational modes and objectives while accounting for uncertain environmental conditions, payload configurations, and possible failures of onboard actuators. Adaptive control has many applications in the underwater domain and can give a robotic diver's assistant the flexibility required to enable joint robot-diver operations. Therefore, two adaptive control system approaches, Model Reference Adaptive Control and $\mathcal{L}_1$ Adaptive Control, are investigated here for heave control of the Tethered, Hovering Autonomous Underwater System.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AUV	Autonomous Underwater Vehicle
CAVR	Center for Autonomous Vehicle Research
DOF	Degrees of Freedom
DVL	Doppler Velocity Log
GPS	Global Positioning System
INS	Inertial Navigation System
$\mathcal{L}_1$ AC	$\mathcal{L}_1$ Adaptive Control
LLS	Linear Least Squares
MRAC	Model Reference Adaptive Control
MRAC	Model Reference Adaptive Control
NPS	Naval Postgraduate School
RDAS	Robotic Diver's Assistant System
ROV	Remotely Operated Vehicle
SISO	Single Input Single Output
SysID	System Identification
THAUS	Tethered Hovering Autonomous Underwater System

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to offer my sincerest thanks to my advisor, Dr. Noel Du Toit, for his guidance and mentorship. His support during my research made my time at the Naval Postgraduate School truly worthwhile. I would also like to thank Dr. Vladimir Dobrokhodov, whose excitement and readiness to share his wealth of knowledge was beyond valuable. Finally, I must thank my family and friends for their constant support. From my home in Texas to the docks of Monterey, wherever I turned, their encouragement kept me motivated.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Research in the field of robots has given increasing prominence to the study of autonomous underwater vehicles. While academic and research applications have long been familiar with robotics, the military and commercial communities are showing increased interest in the capabilities they can offer. This interest is quickly raising demand for robotic platforms than can work closely with a human operator. In particular, the benefit of leveraging robotic platforms to aid divers in the performance of their duties is now evident for diving communities such as researchers, marine engineers, and the military. Of particular importance are the improved efficiency, effectiveness and safety such a system can bring to these underwater operations. Diving operations, occurring in a sensory depriving, isolating, dynamic environment, are inherently dangerous. These operations are demanding not only on the divers themselves, but also on the support crew and their resources.

The Robotic Diver's Assistant System (RDAS) being developed by the Naval Postgraduate School's (NPS) Center for Autonomous Vehicle Research (CAVR) could alleviate some of the workload that burdens a dive team by augmenting human operations and engaging in tasks such as inspection, tool manipulation, navigation, or equipment carrying. These RDAS capabilities have the potential to particularly benefit Navy operators in Salvage, Search and Rescue, or Explosive Ordinance Disposal missions as well as NASA and other civilian communities such as law enforcement, commercial marine engineers, or scientific researchers involved in complex underwater operations. Figure 1. demonstrates the conditions in which a RDAS is likely to operate.

Figure 1. Divers in Close-Quarters Operations at NASA Aquarius Reef Base

The ability to precisely control an RDAS is critical to the success and safety of any underwater mission. The arduous undersea environment subjects the vehicle to many external disturbances such as current and swells. The vehicle will have to overcome these disturbances and remain steady on task. Additionally, the potential for system failures exists that will change the characteristics of the system. An inability to work around or overcome such failures risks the safety of the diver or other systems in their vicinity and potential mission failure. Alternatively, and more commonly, the RDAS will be carrying out missions requiring varying payloads and system configurations (i.e., mid-mission changes), which will change the system's base dynamic characteristics as well. A change in the AUV's payload configuration changes the vehicle's buoyancy, center of mass, and drag. Any variation of these parameters, alone or in combination, diminishes the ability of the RDAS to properly carry out its mission, unless the on board control system is capable enough to sufficiently compensate for each of these parameter variations. As is common with many aircraft control system designs, it may be possible to anticipate many of the different parameter configurations that could potentially occur and then develop control systems configurations optimized to correspond to each individual scenario. This practice, known as "gain scheduling" [1] is feasible when the scenarios are sufficiently foreseeable and limited in scope; however, this is usually not the case for an RDAS that

has to operate in multiple mission capacities. Gain scheduling in a THAUS, under real-world conditions is extremely difficult and burdensome to the point of impracticality. The number of variables affecting the vehicle's dynamics is extensive and difficult to calculate, let alone account for with a predefined gain table.

This research investigates an alternate control system method suitable for application in the set of scenarios in which THAUS parameter variation occurs: adaptive control is an approach in which the control system is able to modify its behavior in response to changes in the target system's inherent dynamics, system response to environmental disturbances, or platform failures. Unlike traditional control approaches, adaptive control systems are able to operate with a consistent design under a variety of configurations without the need to retune control gains and, unlike gain scheduling, no library of control parameters corresponding to anticipated scenarios is required. Adaptive control is most practically used in a system whose process dynamics vary unpredictably or that will be operated in an environment with inconsistent disturbances [1]. Two specific types of adaptive control will be evaluated in this research. Direct Model Reference Adaptive Control (MRAC) forces the behavior of the target system to mimic that of a *reference* model. $\mathcal{L}_1$ Adaptive Control ($\mathcal{L}_1$ AC) uses a predictor model to estimate system states and adapts control efforts to match this predictor system while ensuring that the resulting control signals are physically executable. The appropriate application of these adaptive control systems has the potential to provide the flexible, stable, and precise control required for the successful operation of an RDAS.

B. LITERATURE REVIEW

1. Introduction

As a whole, the control of unmanned vehicles is concerned with the ability to carry out a desired task such as moving to a specified pose with both speed and accuracy. Research continues to address challenges associated with this goal, such as finding optimal control solutions, disturbance rejection, and changing environments. While the problems that have presented themselves are both numerous and broad, those that are

most relevant in this research concern system modeling and system control. These two problems go hand in hand and improvements in one benefit the shortcomings in the other.

2. System Modeling

Many researchers have focused on the problem of developing an accurate system model for underwater platforms. Such a model is beneficial since it permits simulation and controller design using a valid metric. The principal step in the model development process is to define a parametric model that captures the relationships between the various forces and moments that act on a body and the resulting motion (i.e., dynamic and kinematic models). These models are often very complex and require careful consideration of many factors, such as the ability to identify the parameters associated with the selected model. In the case of marine systems, analysis of rigid body dynamics is insufficient to describe a system that is also subject the highly tumultuous forces in the underwater environment. Hydrodynamic effects are also necessarily considered, resulting in a model with over 200 unknowns. Fossen [2] discusses a detailed approach for developing kinematic and dynamic models for generalized underwater vehicles. This generalized model is applied to nonlinear motion of a six degree of freedom (DOF) vehicle and contains components addressing rigid body and added mass, Coriolis effects, damping and restoring forces.

3. System Modeling for THAUS-like Platforms

Yuh [3] presents and evaluates a dynamic model for a six degree of freedom, remotely operated, underwater vehicle used in conjunction with a self-tuning control system. Though he recognizes its limitations, Yuh's model incorporates both rigid body and hydrodynamic forces in its development. This model is a specialization of the model presented by Fossen [2] through the introduction of appropriate assumptions. In previous work at Naval Postgraduate School, Weiss [4] defines a simplified dynamic model for the SeaBotix THAUS based on Fossen's approach and using assumptions similar to Yuh. This model was developed using both rigid body and hydrodynamic analysis and includes added mass, fluid motion, drag, weight and buoyancy, and external force effects. Both Yuh and Weiss are able to simplify their models by making reasonable assumptions

about the vehicle and its operating characteristics, such as shape, composition, symmetry, center and homogeneity of mass, medium density and turbulence, and vehicle speed. Using similar assumptions Caccia et al. [5] reduce the general dynamic equation to a decoupled model structure for a single degree of freedom motion. Streenan [6], a former student at NPS, also effectively implements a simplified, decoupled model for planar motion for investigation in diver tracking. This decoupling allows for identification of hydrodynamic parameters by exciting motion in isolated degrees of freedom, limiting the complexity of the calculations required to solve for those parameters.

4. System Identification

Model learning, also known as system identification (SysID), is an integral step to any adaptive control technique, including MRAC and $\mathcal{L}_1$ AC, that provides the reference and predictor models used in the adaptation processes. The SysID process attempts to identify the individual parameters of the dynamic model, such as drag coefficients, by deliberately exciting a particular vehicle dynamic and then evaluating the response. Two basic categories of system identification techniques exist: offline and online. The offline system identification techniques analyze a complete data set of vehicle motion and selects parameters of the dynamic model that best represent the vehicle motion. Various techniques are offered in texts by Slotine and Li [7] as well as Astrom and Wittenmark [1] and include linear least squares regression, a method for minimizing compounded model error with respect to the data set, and gradient estimators, which minimize step to step error. These techniques are expanded for online application, which can incrementally update parameter estimates as additional data is received. These techniques implicitly assume time invariant parameters, and techniques have been explored to allow estimation of time-varying parameters: forgetting techniques can limit the data set to the most recently available data in order to improve estimation accuracy.

5. SysID for THAUS-like Systems

Weiss [4] applies these and other techniques in the system identification of the SeaBotix THAUS. Yang et al. [8] suggest methods for model definition and parameter identification of a complexly shaped semi-AUV, the CISCREA. Their research borrows

primarily from Fossen [2] in the development of their dynamic models and then focuses heavily on the identification of hydrodynamic parameters via multiple software based computational fluid dynamics methods. Their methods avoid the use of expensive experimental analysis such as tow tanks, yet produce adequately accurate results. On the other hand, Caccia et al. [5] experimentally derive the dynamic coefficients on the ROME underwater vehicle. Their experiments carefully measure the vehicle kinematics resultant from a nominal applied force. This data is then used to create a “best fit” model.

6. Control of THAUS-like Platforms

AUVs that operate in the ocean environment are frequently subjected to uncertainties due to nonlinear hydrodynamics, sea state disturbances, or parameter variation. As a result, a once suitable system model, with accurately identified parameters, may no longer be appropriate. Classical control systems, such as proportional, derivative and integral controllers presented by Nise [9], while suitable for static, well defined systems, are unable to properly control systems with variable parameters. In this common instance, more advanced and robust control methods such as adaptive control are rightfully considered. The performance of an adaptive control system is evaluated similar to classical systems. Three design objectives in modern control theory are robustness (the ability function properly with uncertain or inaccurate parameters,) stability (the ability to converge to a steady, predictable behavior,) and rate of response. In the presence of time-varying parameters and/or nonlinear system responses, classical control techniques fail to perform as required while adaptive techniques prove capable.

7. Adaptive Control

The adaptive control methods investigated in current research have potential to provide the required robustness, stability and response speed to the RDAS even in the presence of uncertainties and time variant parameters. Several of the adaptive methods are presented in texts by Ioannou and Sun [10], Slotine and Li [7], and Astrom and Wittenmark [1], including MRAC (direct and indirect), and self-tuning control. These controllers ideally allow the system to operate successfully with limited a priori

knowledge of the system model and environment it is operating in. The origin for research into adaptive systems is rooted in the development of aircraft control systems operating in a wide range of flight conditions [11]. Other research on adaptive control system implementation such as those by Slotine and Li [12] and Nicosia and Tomei [13] focused on industrial applications involving robotic manipulator arms carrying various loads. The application of adaptive techniques has since widely expanded. Adaptive techniques have more recently come to be applied to ground vehicles and aircraft systems and now to marine vehicles as well.

8. Adaptive Control in Maritime Systems

Work by Fossen and Sagatun [14] demonstrated the successful application of adaptive techniques developed for robotic manipulators and spacecraft to the NEROV, nonlinear, underwater robotic system. Maalouf et al. [15] demonstrated the more robust performance of an adaptive controller versus a classical proportional-derivative controller in implementation on the Triton-PR submarine. Many variations of these controllers exist as well; for example, Cheah et al. [16] have developed an adaptive Jacobian controller, a type of indirect MRAC, designed to address both kinematic and dynamic uncertainties. Further modification can be made to adaptive controllers that are designed to increase the robustness of the systems in the presence of high adaptation gains. Sanei et al. [17] compare two such modifications: the dead-zone modification, which establishes a bounded range about the reference command where controller parameter adjustment does not occur, and the Projection Operator controller, which constrains the controller parameters and limits their rate of change. Another modifier, the σ-modification, utilized by Wen et al. [18], acts as an adaptation dampener and prevents unstable changes to controller parameters. These modification and variations attempt to address the issue of coupling between rate of adaptation and robustness. Because they are coupled, prioritizing one metric requires sacrificing performance in the other. This shortcoming requires precise tuning of controller parameters and careful selection of reference models.

9. $\mathcal{L}_1$ Adaptive Control

While the majority of adaptive controllers seek to increase the speed of adaptation and steady state tracking performance, these improvements often come at the cost of losing robustness. A more recently developed adaptive controller, $\mathcal{L}_1$AC has been introduced by Hovakimyan and Cao [19] to address this coupling between rate of adaptation and robustness. A useful comparison of the robustness of several adaptive controllers (direct and indirect MRAC, direct MRAC with the σ-modification) with $\mathcal{L}_1$AC is provided. The benefit that $\mathcal{L}_1$AC provides is a novel decoupling of robustness (i.e., stable, accurate performance), and adaptation and convergence speed. The $\mathcal{L}_1$AC architecture was recently implemented with success by Maalouf et al. [20] on an AC-ROV underwater vehicle. This research shows the $\mathcal{L}_1$AC to be a viable candidate for AUV control.

C. RESEARCH OBJECTIVES AND SCOPE

The goal of this research is the development of an adaptive control system that will enable an AUV to operate with sufficient precision to conduct joint human-robot missions as an RDAS. The capabilities of this control system are characterized by:

- Precise motion with disturbance rejection
- Precise control with configuration changes
- Dynamic stabilization for task execution

To realize these goals, the following tasks were executed:

- Selection of a simplified, decoupled dynamic model for use as a reference model and in simulation.
- Development of a MRAC and accompanying modifications to increase robustness.
- Development of a $\mathcal{L}_1$AC.
- Implementation and experimentation with these controllers for evaluation and comparison on the THAUS robotic platform.

The thesis is structured as follows:

Chapter II describes the test platform being used, the SeaBotix vLBV300. The model used on this vehicle is also discussed, starting with the appropriate kinematic and dynamic equations of motion and then outlining the modeling assumptions to obtain a simplified model.

Chapter III focuses on the development of MRAC and the theory supporting it. An overview of the controller's expected capabilities and limitations follows. The chapter also discusses the design of simulations and experiments.

Chapter IV similarly discusses the development of the $\mathcal{L}_1$AC, its supporting theory and expected performance. Simulation and experimentation design are also explained.

Chapter V focuses on the experimental results of the implementation of the controllers from Chapters III and IV in simulation and on the vehicle. This chapter includes discussion of the individual performance of the controllers as well as their comparative performance.

Chapter VI then presents the conclusions of this research: performance advantages and limitations, reevaluation of goals, and recommendations for refinement, and future work. This chapter also provides a brief summary of the key points of the research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. AUV PLANT MODELING

A. SEABOTIX VLBV300 PLATFORM

The vLBV300, a mini-Remotely Operated Vehicle (ROV), is commercially available and produced by SeaBotix, Inc. of San Diego, California (Figure 2.). The vLBV300 uses an open frame construction that makes the vehicle extremely customizable and capable of incorporating a wide range of sensors and tools onto its chassis with relative ease. Additionally, due to its vectoring thrust and dynamically stable architecture, the vehicle is agile and steady, a necessary trait for robot-diver operations. The ROV has the key ability to move laterally, making it especially well-suited for precise positioning. This versatility makes it a viable candidate for a large selection of missions. The unit is built to be rugged as well, a requirement for any military operation. Because of these advantages, the U.S. Navy's diving and salvage teams use these vehicles in the fleet. SeaBotix designed the vLBV300 with such utility in mind and recommends the unit for use in work on offshore oil and gas structures, coastal and inshore surveys, maritime security, and scientific research.

Figure 2. The SeaBotix vLBV300 ROV

1. System Configurations

In its base configuration the vehicle is operated by joystick via a control interface. In comparison, the Tethered Hovering Autonomous Underwater System (THAUS) is an augmented and modified vLBV300, which can be operated in a variety of control modes,

including semi-autonomous and fully autonomous operations. In the course of this research the vehicle is operated in an autonomous mode. The commands are relayed to the vehicle via a tether. All sensor and command data is sent through this conduit, which also serves as the power connection from a surface stationed AC power source. The tether also supports live video feeds, a critical capability in many applications of interest to the Naval dive communities. The vehicle operates freely in 5-DOF but only four of these channels are normally considered: surge, sway, heave, and yaw. Roll control is limited, and pitch is not controllable, due to thruster configuration. These four channels provide sufficient control and freedom of movement and are common in hovering class underwater vehicles. A total of six vectored, 100 mm diameter propellers provide thrust. The forward horizontal thrusters are offset at an angle of 35° from centerline. Aft horizontal thrusters are offset at 45°. These vectored offsets allow the vehicle to effectively maneuver in the surge, sway, and yaw directions. Vertical thrusters are offset by 18° from center allowing the vehicle to maneuver in the heave and roll directions; however, roll motion is largely undesired and is negated by the buoyant force / weight righting moment arm. Figure 3. depicts the vectored thrust configuration. The vehicle is capable of speeds in excess of 3 knots at full power.

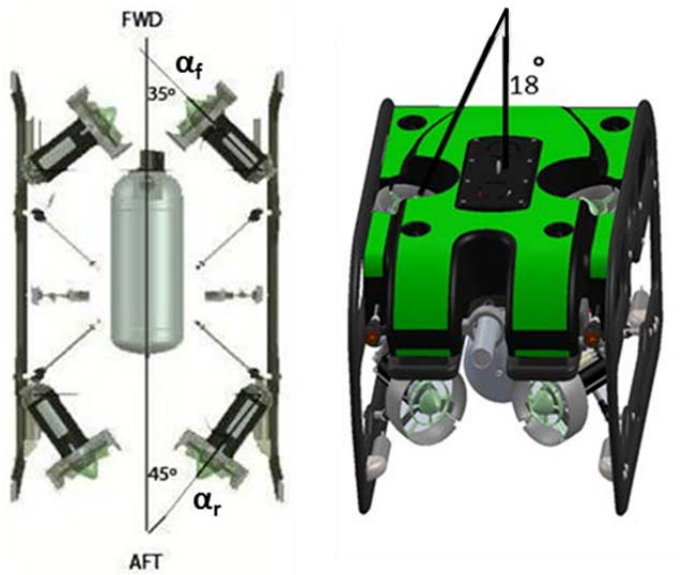

Figure 3. SeaBotix vLBV300 Vectored Thrust Configuration, after [4], [21]

The nominal vehicle specifications, listed in Table 1, are provided by SeaBotix, Inc. Some of these specifications will change based on CAVR's specific sensor payload and will be listed in the next section.

Depth Rating	300 m
Length	625 mm
Width	390 mm
Height	390 mm
Diagonal	551 mm
Weight in Air	18.1 kg
Maximum Forward Thrust	22.5 kgf
Maximum Lateral Thrust	15.2 kgf
Maximum Vertical Thrust	9 kgf

Table 1. SeaBotix vLBV300 System Specifications, from [21]

2. Onboard Sensors

The vLBV300 ships with an integrated camera and lighting system on a tilt unit that feeds real-time video to the surface station. Several other aftermarket sensors and accessories have been added to the system. A BlueView P450-45, 2D imaging, forward looking sonar system was added, weighing 2.6 kg in air and 0.56 kg in water [22], and gives the vehicle the ability to supplement search and mapping missions. The expanded sensor payload also includes a combination of Inertial Navigation System (INS) and Global Positioning System (GPS) provided by Greensea Systems, Inc. weighing 8 kg in air and is nearly neutrally buoyant in water [23]. The INS incorporates a separate Doppler Velocity Log (DVL) by Teledyne RD Instruments Inc.[24], weighing 4.3 kg in air and 0.8 kg in water. The combined weight of the fully integrated system, including extra ballasting weight is 33 kg in air and neutrally buoyant in water in the nominal configuration. These systems grant the vehicle position accuracy necessary to operate in the vicinity of divers. System specific accuracies for the INS and DVL are summarized in Table 2. In addition to the sensor payload, the THAUS is also equipped with a claw like, actuated manipulator. Integration of this payload can be seen in Figure 4.

INS / GPS	
Position Accuracy	0.1% distance traveled RMS
	1.0 m / min drift
Depth Accuracy	0.01%
Depth Range	0 to 6000 m
Velocity Accuracy	0.2% ±.001 m/s
Velocity Range	±10 m/s
DVL	
Velocity Accuracy	±0.3% ±0.2 cm/s
Velocity Range	±9 m/s
Altitude Range	0.5 to 81 m

Table 2. Summary of Sensor Accuracies, from [23], [24]

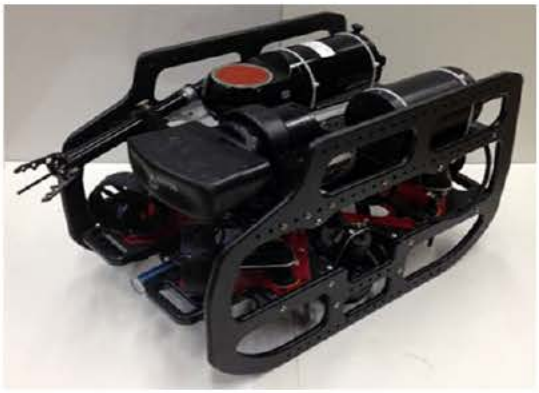

Figure 4. SeaBotix vLBV300 (Inverted) with Customized Sensor Payload

B. DYNAMIC EQUATIONS OF MOTION DEVELOPMENT

Kinematic and dynamic equations of motion are required to describe the motion of an AUV and the forces and moments that influence that motion. These general equations are then used to develop a specialized model for the THAUS, so that we may predict the vehicle's response to different stimuli, be they commanded or disturbances. These equations are derived from work by Fossen [2] using notation standardized by the Society of Naval Architects and Marine Engineers, and are summarized below. These generic equations were simplified by Weiss [4] in prior work for application to the THAUS, resulting in a simplified model (based on relevant assumptions). This model is adopted here for use in the development of the control systems in this research. The derivation of this model, with appropriate assumptions, is summarized below.

1. Coordinate System and State Vectors

First, it is necessary to define the associated coordinate systems to be used. A right-handed, body-fixed frame is established: +*x* pointing forward (surge), +*y* pointing right sway), and +*z* pointing down (heave) as in Figure 5. Rotation about the z-axis is yaw, rotation about the y-axis is pitch, and rotation about the x-axis is roll.

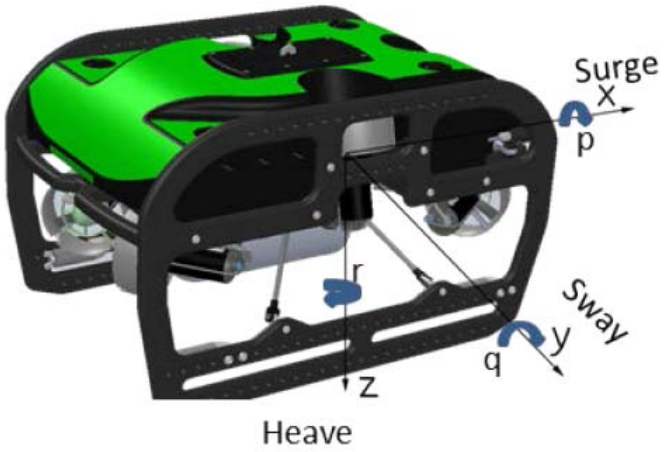

Figure 5. SeaBotix vLBV300 Body-Fixed Coordinate Frame, from [4]

The linear and angular motion velocities, are represented by vector, υ

$$\upsilon = [u, v, w, p, q, r]^T \tag{1}$$

where *u* is the surge direction velocity, *v* is the sway direction velocity, *w* is the heave direction velocity, *p* is the angular rate about the body frame x-axis (roll rate), *q* is the angular rate about the body frame y-axis (pitch rate), and *r* is the angular rate about the body frame z-axis (yaw rate). Velocities and angular rates are represented in units of meters per second or radians per second, respectively.

The pose of the vehicle is represented by vector, η

$$\eta = [x, y, z, \phi, \theta, \psi]^T \tag{2}$$

where *x, y,* and *z* are the vehicle's coordinates with respect to an inertial reference frame, and $\phi, \theta, \text{and } \psi$ are Euler angels determined with the "*zyx*" convention with rotating frames. Positions and angles are presented in units of meters and radians, respectively.

2. Propulsion Model

In addition to using the joystick, the AUV can be controlled with low level, program based, Pulse Width Modulation (PWM) commands directly to the individual thrusters. The PWM commands can range from -102 to 102 and are related to thruster RPM, which translates to thrust (Newtons) according to the equation,

$$Thrust = 0.006736PWM^2 - 0.03366PWM + 0.0684 \tag{3}$$

Thrust is then further translated to motion according to the position and vectoring of each individual thruster. In the equations that follow, *F* represents a forward thruster, *A*, an after thruster, *P*, a port thruster, *S*, a starboard thruster, and *V*, a vertical thruster. In combination, FP refers to the thrust associated with the *forward-port* thruster, etc. Additionally, α_f is the offset angle for forward thrusters, α_r is the offset angle for aft thrusters, and β is the offset angle for vertical thrusters as shown in Figure 3. X_{prop} (Equation (4)) is the combined generated force from the individual thrusters in the surge direction; Y_{prop} (Equation (5)) is the total thrust in the sway direction, and Z_{prop} (Equation (6)) is the total thrust in the heave direction. N_{prop} (Equation (7)) is the total moment generated about the body frame z-axis. Equations for the remaining degrees of freedom ($K_{prop,}$ moment about the x-axis, and $M_{prop,}$ motion about the y-axis) are omitted since these channels are not used in this research. Finally, LH_1 and LH_2 are moment arms referring to the distance between the centroid and the aft-forward and port-starboard pairs of propellers, respectively. LV is the moment arm between vertical propellers and the centroid [4].

$$X_{prop} = (AP\, c\alpha_r + FP\, c\alpha_f) + (AS\, c\alpha_r + FS\, c\alpha_f) \tag{4}$$

$$Y_{prop} = (-FS\, \mathrm{s}\alpha_f + AP\, \mathrm{s}\alpha_r) - (-FP\, \mathrm{s}\alpha_f + AP\, \mathrm{s}\alpha_r) - (VP - VS)\, \mathrm{s}\beta \tag{5}$$

$$Z_{prop} = (VP + VS)\, \mathrm{c}\beta \tag{6}$$

$$
\begin{aligned}
N_{prop} = {} & (AP\,c\alpha_r + FP\,c\alpha_f)\,LH_1 + \\
& (AS\,c\alpha_r + FS\,c\alpha_f)\,LH_1 + \\
& (AP\,s\alpha_r + FP\,s\alpha_f)\,LH_2 + \\
& (AS\,s\alpha_r + FS\,s\alpha_f)\,LH_2
\end{aligned}
\tag{7}
$$

Note: for simplicity of notation $s\bullet = sin(\bullet)$, and $c\bullet = cos(\bullet)$.

3. Assumptions and Simplified Model

Modeling the non-linear dynamics of the THAUS requires that the following two equations be solved for υ and η, as per Fossen [2]. The first represents the sum of forces and moments, τ, on the vehicle. *M* is the rigid body and added mass term, $C(\upsilon)$ is the rigid body and added mass Coriolis effects, $D(\upsilon)$ is the damping term, and $g(\eta)$ is the reactionary force term. The second relates vehicle pose dynamics to body velocities, angular rates, and pose. $J(\upsilon)$ is a 6x6 rotation matrix defined by the Euler angles.

$$
M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{8}
$$

$$
\dot{\eta} = J(\eta)\upsilon \tag{9}
$$

Solving Equations (8) and (9) for (1) and (2) can be simplified by making some basic assumptions about the characteristics of the vLBV300:

- The tether has no effect on vehicle motion or trim.
- The x and y position of the center of gravity, center of buoyancy, and the body-frame are coincident. The z-position may be offset
- The vehicle has a uniform mass distribution.
- Three planes of symmetry exist on the vehicle.
- All motion occurs at low speed, fully submerged.
- All flow is completely turbulent.

The application of these assumptions yields the following coupled, simplified, dynamic equations of motion for the THAUS AUV [4]:

$$
\begin{aligned}
X_{prop} &= (m - X_{\dot{u}})\dot{u} - (m - Y_{\dot{v}})vr + (m - Z_{\dot{w}})wq - X_{u|u|}u\,|u| + (W - B)\sin\theta \\
Y_{prop} &= (m - Y_{\dot{v}})\dot{v} - (m - Z_{\dot{w}})wp + (m - X_{\dot{u}})ur - Y_{v|v|}v|v| - (W - B)\cos\theta\sin\phi \\
Z_{prop} &= (m - Z_{\dot{w}})\dot{w} - (m - X_{\dot{u}})uq + (m - Y_{\dot{v}})vp - Z_{w|w|}w|w| - (W - B)\cos\theta\cos\phi \\
K_{prop} &= (I_X - K_{\dot{p}})\dot{p} + (Y_{\dot{v}} - Z_{\dot{w}})wv + (I_Z - I_Y + N_{\dot{r}} + M_{\dot{q}})rq \\
&\quad -K_{p|p|}p|p| + (z_g W - z_b B)\cos\theta\sin\phi \\
M_{prop} &= (I_Y - M_{\dot{q}})\dot{q} + (Z_{\dot{w}} - X_{\dot{u}})uw + (I_X - I_Z + N_{\dot{r}} - K_{\dot{p}})rp \\
&\quad -M_{q|q|}q|q| + (z_g W - z_b B)\sin\theta \\
N_{prop} &= (I_Z - N_{\dot{r}})\dot{r} + (X_{\dot{u}} - Y_{\dot{v}})uv + (I_Y - I_X + K_{\dot{p}} - M_{\dot{q}})rp \\
&\quad -N_{r|r|}r|r|
\end{aligned}
\tag{10}
$$

The added mass terms are contained in $M_A(\upsilon) = -diag([X_{\dot{u}}\ Y_{\dot{v}}\ Z_{\dot{w}}\ K_{\dot{p}}\ M_{\dot{q}}\ N_{\dot{r}}])$, Coriolis terms are defined in terms of added mass variables, the dampening terms are contained in $D(\upsilon) = -diag([X_{u|u|}\ Y_{v|v|}\ Z_{w|w|}\ K_{p|p|}\ M_{q|q|}\ N_{r|r|}])$, the $I_X, I_Y, \text{and}\, I_Z$ terms are the first moments of inertia and z_g and z_b are the z-direction, Cartesian, body coordinates of the center of gravity and center of buoyancy, respectively. Additionally, it is assumed that the vehicle will remain close to level; therefore, $\theta \approx 0$ and $\phi \approx 0$. This reduces the restoring forces to $g(\eta) = [0, 0, -(W - B), 0, 0, 0]^T$.

Within the scope of this research, only the heave channel motion is considered. This focus permits isolation of the heave direction dynamic equations of motion. Since the heave channel can be excited independently, the other channels and coupling terms can be ignored (they are not excited). The application of these final assumptions yields this simplified heave motion equation,

$$
Z_{prop} = (m - Z_{\dot{w}})\dot{w} - Z_{w|w|}w|w| - g_z(\eta) \tag{11}
$$

where *m* is the system mass, $Z_{\dot{w}}$ is the added mass, and $Z_{w|w|}$ is the quadratic dampening term, and $g_z(\eta)$ is the restoring force term equal to $Weight - Buoyancy$. This equation will be used later in Chapter III to represent the reference model in both simulation and experimental trials.

4. System Identification

The final step in developing an acceptable system model that can be used in a simulator and during the controller design process is to solve for the unknown parameters: the added mass and dampening terms. While we expect these parameters to change in actual operations, identifying the terms for a nominal system allows us to use this model as a reference for an MRAC system, as a predictor for the $\mathcal{L}_1$ AC system, and for performance analysis. The Linear Least Squares (LLS) method is used to estimate the unknown parameters [1].

Equation (11) can be rearranged as

$$\dot{w} = \frac{Z_{w|w|}}{m - Z_{\dot{w}}} w|w| + \frac{1}{m - Z_{\dot{w}}}(Z_{prop} + g_z(\eta)) \tag{12}$$

and then written in regression form

$$y = \Phi * \Theta^T \tag{13}$$

where $y = \dot{w}$ is the first time derivative of z-axis velocity, $\Phi = [w|w|, (Z_{prop} + g_z(\eta))]$ is measured experimentally, and $\Theta = [\frac{Z_w|w|}{m - Z_{\dot{w}}} \quad \frac{1}{m - Z_{\dot{w}}}]$ is the set of unknown parameters. The LLS algorithm seeks to minimize the square of the error between the measured and modeled system response ($\varepsilon_i = y_i - \Phi_i \Theta_i^T$ and $i = 1, 2, ..n$ observations) with respect to Θ, according to the cost function, $V(\Theta, n)$:

$$V(\Theta, n) = \frac{1}{2}\sum_{i=1}^{n} \varepsilon_i^2 \tag{14}$$

This cost function is minimized by

$$\hat{\Theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{15}$$

which is also the estimated solution for the system's unknown parameters.

5. Experimental SysID

A simple experiment was designed to estimate the parameters present in the model in Equation (12), Θ. Starting at approximately 0 m/s, a step command was passed to the THAUS, equating to a percentage of maximum force. The THAUS would then accelerate and settle to a new steady state velocity (open-loop control). Measured velocity data was differentiated to obtain $\dot{w}$. This data allowed for calculations in terms of Equation (13), and then for calculating a solution via the LLS method in Equation (15) . This process was repeated for 3 z-force step inputs: 30%, 50%, and 80% of maximum with the THAUS in a neutrally buoyant condition. After calculating the unknown terms, the generated models, as well as an average model, were compared to measured data to find the "best fit" model. The model selected corresponds to the 30% step input data. This model provides the best fit to all data sets, especially to the 30% step data, as expected. Additionally, this model is favored because the THAUS will more commonly be operated at low speeds with z velocity in a range of less than 50% of maximum. The results of data are shown in Figure 6. The selected model is the thick red line. The experimental, measured data is the thick blue line. The selected model equates to a solution where $\hat{\Theta} = [-0.4645 \;\; 0.0036]$. The model parameters calculated from this identification are: $Z_{\dot{w}} = -244.3629$ and $Z_{w|w|} = -128.8485$ using a vehicle mass of 33 kg. The parameters calculated for 50% force are: $Z_{\dot{w}} = -267.9320$ and $Z_{w|w|} = -111.9450$. The parameters calculated for 80% force are: $Z_{\dot{w}} = -281.3205$ and $Z_{w|w|} = -103.4917$.

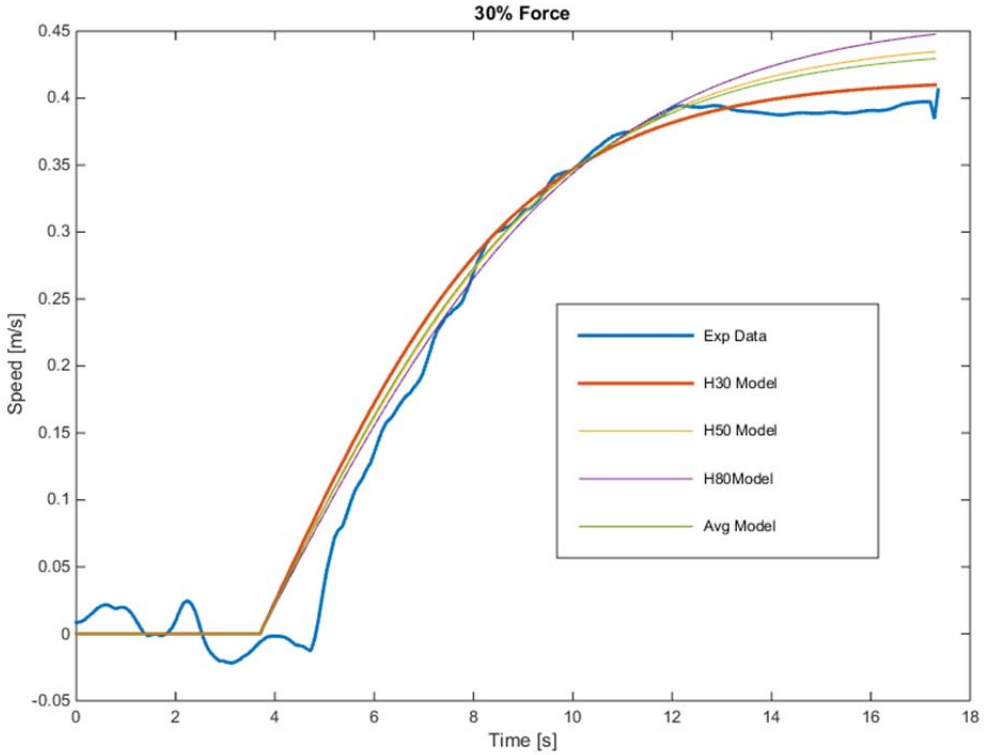

Figure 6. Experimental System Identification Best Fit Model Results

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODEL REFERENCE ADAPTIVE CONTROL SYSTEM

A. DEVELOPMENT

1. Overview

A direct model reference adaptive control system is generally composed of four primary parts. The *plant* is the system to be controlled, that often contains unknown, unmodeled, or incompletely modeled dynamic parameters; however, the basic structure of the dynamic model is assumed to be known. The *reference model* is a system with a known dynamic model, whose response to a command signal specifies the desired behavior of the plant. In order to be useful, the reference model's performance must also be dynamically achievable by the plant. The *controller* sends a modified command signal to the plant and may contain a number of parameters that are allowed to vary. These parameters are changed in order to achieve asymptotic tracking convergence between the plant and reference model. These control parameters are changed according to an *adaptation law*, which takes input from the error between the plant and the model reference. Ideally, as the control parameters converge to a solution that satisfies the dynamic equation, tracking error will trend toward zero. This MRAC architecture is *direct* because the adaptation mechanism estimates and adjusts the controller parameters, vice estimating the actual unknown parameters of the plant and then calculating the controller parameters from these estimates. Figure 7. illustrates the general architecture of a direct MRAC system [7], [11].

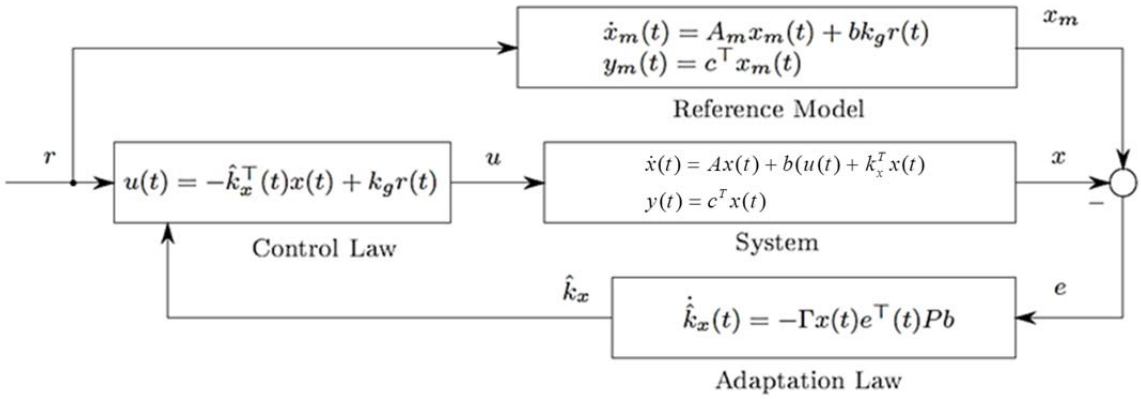

Figure 7. Direct MRAC General Architecture, after [25]

2. System Plant

Implementation of MRAC depends on knowledge of the dynamic model of the system, derived from historical principles. This parametric model determines the exact implementation used. The plant in an MRAC system can be expressed generally as,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\Lambda(u(t) + f(t, x(t))), \quad x(0) = x_0 \\ y(t) &= C^T x(t) \end{aligned} \tag{16}$$

where $x(t) \in \mathbb{R}^n$ is the system state and $u(t) \in \mathbb{R}^m$ is the control input; $x_0 \in \mathbb{R}$ is the state initial condition, known to be in an arbitrarily large, but bounded set; $A \in \mathbb{R}^{n \times n}$, representing the plant's dynamics, and $\Lambda > 0 \in \mathbb{R}^{m \times m}$, the diagonal control efficiency matrix, contain unknown constant parameters, while $B \in \mathbb{R}^{n \times m}$, the control matrix, is assumed to be known. System dynamics contain nonlinear terms, $f(t, x(t))$. From the parametric model, the general structure of these nonlinear terms is known. As a consequence, the function, $f(t, x(t))$ can be written in terms of a known basis function, $\Phi(x(t))$, and unknown coefficients Θ such that $f(\mathrm{t}, x(t)) = \Theta^T \Phi(x(t))$. This is known as "matched" uncertainty. Approaches exist to account for unmatched uncertainty as well (e.g., using radical basis functions, etc.) but these extensions are not considered here [11]. These assumptions allow Equation (18) to be rewritten from MRAC as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\Lambda(u(t) + \Theta^T \Phi(x(t))) \\ y(t) &= C^T x(t) \end{aligned} \tag{17}$$

3. Reference Model

A stable reference model is specified. The reference model parameters are selected such that the reference model represents the ideal, achievable system behavior in response to bounded command input. The reference model must be achievable in that the system it represents must be capable of responding in a similar way. The desired response of this model is typically obtained using simpler control methods such as designing a feedback law that results in the desired closed loop behavior. This equation describes the dynamic behavior of the system as a first order differential equation:

$$\begin{aligned} \dot{x}_m(t) &= A_m x_m(t) + B_m r(t) \\ \mathrm{y}_m(t) &= C^T{}_m x_m(t) \end{aligned} \tag{18}$$

where $A_m \in \mathbb{R}^{n \times n}$ is Hurwitz, $B_m \in \mathbb{R}^{n \times m}$, and $r(t) \in \mathbb{R}^m$ is the time dependent, bounded, reference command input. A_m is chosen to best express the desired system dynamics. B_m is calculated according to the equation, $B_m = B k_g$,where $k_g = \dfrac{-1}{C^T A_m^{-1} B}$. This calculation ensures steady state tracking error for a step function is zero and is derived in Appendix C.

It will be shown in Section B that the dynamics obtained in the previous chapter can be written in this form. Using modeled dynamics as the ideal basis establishes the reference model as a system with achievable performance with a nominal configuration [11].

4. Control Law

Once a reference system has been defined, a control law, $u(t)$ is sought that results in asymptotic tracking of x with respect to x_m; that is, the tracking error $e(t) = x(t) - x_m(t)$ tends to zero as $t \to \infty$. The system must be able to track the reference

model even in the presence of the unknown parameters, $A, \Lambda, \text{and } \Theta$. The ideal control solution for achieving this performance, using a *feedback + feedforward* architecture, is

$$u_{ideal}(t) = K_x^T x(t) + K_r^T r(t) - \Theta^T \Phi(x(t)) \tag{19}$$

which, when substituted into Equation (19), becomes

$$\dot{x}(t) = (A + B\Lambda K_x^T(t))x(t) + B\Lambda K_r^T(t)r(t) \tag{20}$$

The ideal controller parameters, $K_x^T(t) \text{ and } K_r^T(t)$, can then be calculated by solving Equation (24), known as the matching conditions:

$$A + B\Lambda K_x^T(t) = A_m \qquad B\Lambda K_r^T(t) = B k_g = B_m \tag{21}$$

These ideals cannot be attained, however, since $A, \Lambda, \text{and } \Theta$ are unknown. Instead estimates of the quantities are used. In indirect MRAC, $A, \Lambda, \text{and } \Theta$ are estimated instead and then used to calculate $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$. Direct MRAC estimates the controller parameters directly. The requirement to persistently excite the system dynamics for indirect MRAC makes direct MRAC more suitable for the THAUS where stability is very important. In the latter case, the controller parameters do not necessarily have to converge to their ideal values to drive tracking error to zero [7]. The final form of the control signal is then,

$$u(t) = \hat{K}_x^T(t)x(t) + \hat{K}_r^T(t) - \hat{\Theta}^T(t)\Phi(x(t)) \tag{22}$$

and after substituting Equation (25)into Equation (19), the system dynamics become

$$\dot{x}(t) = (A + B\Lambda \hat{K}_x^T(t))x(t) + B\Lambda(\hat{K}_r^T(t)r(t) - (\hat{\Theta}(t) - \Theta)^T \Phi(x(t))) \tag{23}$$

Equation (20) can be subtracted from Equation (26) and combined with the matching conditions to produce the closed-loop tracking error dynamics:

$$\dot{e}(t) = A_m e(t) + B\Lambda[\Delta K_x(t)x(t) + \Delta K_r(t)r(t) - \Delta\Theta^T(t)\Phi(x(t))] \tag{24}$$

where $\Delta K_x(t) = \hat{K}_x(t) - K_x$, $\Delta K_r(t) = \hat{K}_r(t) - K_r$, and $\Delta\Theta(t) = \hat{\Theta}(t) - \Theta$ are the parameter estimation errors [11].

5. Adaptation Law

The controller parameters $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$ are estimated using adaptation laws. Estimating correct controller parameters will cancel out the unknown plant gains leaving only the reference model dynamics to be expressed by the plant, as shown in Equation (26), thus leading to zero tracking error. Essentially, once $\Delta K_x(t) \to 0$, $\Delta K_r(t) \to 0$, and $\Delta\Theta(t) \to 0$, steady state tracking error tends to zero globally, uniformly, and asymptotically: $\lim_{t\to\infty} \| e(t) \| = 0$, as proven by Theorem 9.2 in [11].

The three unknowns controller parameters, $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$, are calculated with the equations,

$$\begin{aligned} \dot{\hat{K}}_x(t) &= -\Gamma_x\, x(t)\, e(t)\, PB\, \mathrm{sgn}(\Lambda) \\ \dot{\hat{K}}_r(t) &= -\Gamma_r\, r(t)\, e(t)\, PB\, \mathrm{sgn}(\Lambda) \\ \dot{\hat{\Theta}}(t) &= \Gamma_\Theta\, \Phi(x(t))\, e(t)\, PB\, \mathrm{sgn}(\Lambda) \end{aligned} \tag{25}$$

where Γ_x, Γ_r, and Γ_Θ are the adaptation gains that determine the adaptation rates. P is the solution to the Lyapunov equation, $A_m^T P + P A_m = -Q$, and $Q \in \mathbb{R}^{n\times n}$ is an arbitrary positive definite matrix (e.g., identity). These three adaptation laws incrementally adjust the control gains, improving on the solutions to the dynamic equations and forcing the plant to track the behavior of reference model in response to the reference input, $r(t)$ [11].

6. Modifications

In the presence of unmatched disturbances, the robustness of a system can be reduced as the controller attempts to converge to a tracking solution. A phenomenon known as “parameter drift” occurs. Because the control law cannot adequately compensate for all of the dynamics of the system, the controller parameters, $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\Theta}(t)$ may continue to grow unbounded while the adaptation law seeks

a solution, even while $e(t)$ remains bounded. The occurrence of parameter drift is problematic in that it can result in an overall increase in control effort and the presence of high frequency content in the control signal. Two modifications are made to the adaptive laws that increase the robustness of the system. The first is the *dead zone* modification that changes the adaptation laws in Equation (25) slightly:

$$\begin{aligned}\dot{\hat{K}}_x(t) &= -\Gamma_x\, x(t)\,\tilde{e}(t)\,PB\,\mathrm{sgn}(\Lambda)\\ \dot{\hat{K}}_r(t) &= -\Gamma_r\, r(t)\,\tilde{e}(t)\,PB\,\mathrm{sgn}(\Lambda)\\ \dot{\hat{\Theta}}(t) &= \Gamma_\Theta\, \Phi(x(t))\,\tilde{e}(t)\,PB\,\mathrm{sgn}(\Lambda)\end{aligned} \tag{26}$$

where $\tilde{e}(t)$ is defined conditionally according to the relationship between $e(t)$ the largest permissible error, δ:

$$\tilde{e}(t) = \begin{cases} e(t) & if \quad \| e(t) \| \geq \delta \\ 0 & if \quad \| e(t) \| < \delta \end{cases} \tag{27}$$

This modification helps to reduce oscillations in the control gains and the control signal by eliminating the adaptation when tracking error is sufficiently small, while also ensuring parameter convergence. It requires acceptance of a non-zero steady state tracking error; however, this error is typically tolerably small and can be made arbitrarily small through careful selection of δ [10].

The second modification that was used is the Projection Operator. The Projection Operator allows the MRAC system to maintain robustness in the presence of uncertain dynamics such as unmodeled parameters, while still allowing fast adaptation and system stability. The Projection Operator, sometimes called an "anti-windup" modifier, continues to modify the adaptive parameters while maintaining uniform boundedness of the parameters. The parameter drift phenomenon is halted by the Projection Operator at a predefined limit. For an unknown parameter, θ and its estimate, $\hat{\theta}$, established by the adaptation law, $\dot{\hat{\theta}} = y$, a bound, θ_{max}, is established such that $\mathrm{Proj}(\theta, y)$ operation maintains the resultant θ with the bounds of θ_{max}. To achieve this, a function of θ, and

its gradient are defined in Equation (28). This function includes a projection tolerance, ε, adding a buffer to the estimate and scaling the function.

$$f(\theta) = \frac{(1+\varepsilon)\,\|\theta\|^2 - \theta_{max}^2}{\varepsilon\,\theta_{max}^2}$$
$$\nabla f(\theta) = \frac{2\,\theta(1+\varepsilon)}{\varepsilon\,\theta_{max}^2} \tag{28}$$

In the event that the adaptation law, *y*, results in a parameter estimate, $\hat{\theta}$, that lies outside the defined bounds, θ_{max}, the estimate is redirected to within parameter bounds, thus limiting the growth of $\hat{\theta}$, and maintaining robustness. This limitation is achieved by the following operation, which redefines $\dot{\hat{\theta}}$:

$$\dot{\hat{\theta}} = \text{Proj}(\hat{\theta}, y) = \begin{cases} y - \|\nabla f(\theta)\|^2 \; f(\theta)\, y & \text{if}, \quad [f(\theta) > 0 \text{ and } (y \nabla f(\theta)) > 0] \\ y & \text{otherwise} \end{cases} \tag{29}$$

Figure 8. illustrates the Projection Operator. An estimate that would result in a vector leaving the bounded area is then redirected along the boundary, θ_{max}. Similarly, if an estimate vector results in a vector ending within the projection zone, the Projection Operator redirects the vector to remain within the θ_{max} bounds.

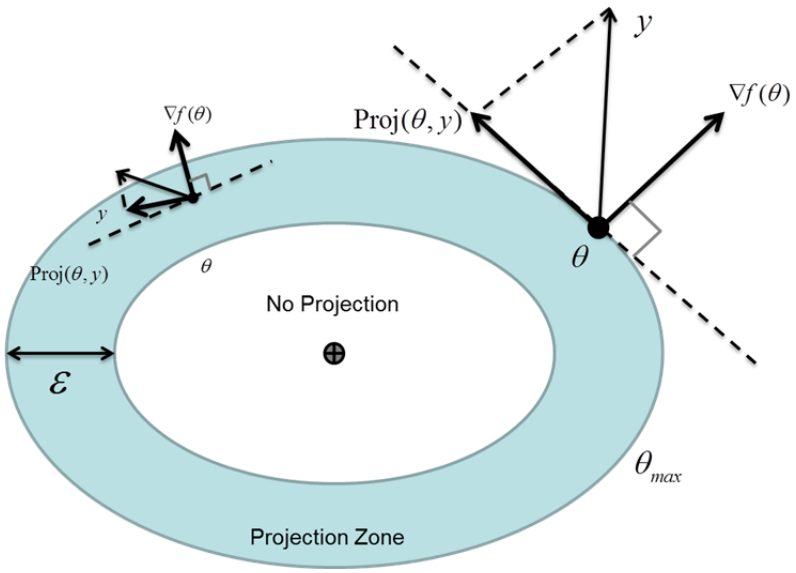

Figure 8. The Projection Operator, from [11]

Subsequently, the Projection Operator is applied to the adaptation laws from Equation (26) [11], [25].

$$\begin{aligned}
\dot{\hat{K}}_x(t) &= \mathrm{Proj}(\hat{K}_x(t), -\Gamma_x\, x(t)\,\tilde{e}(t)\,PB\,\mathrm{sgn}(\Lambda)) \\
\dot{\hat{K}}_r(t) &= \mathrm{Proj}(\hat{K}_r(t), -\Gamma_r\, r(t)\,\tilde{e}(t)PB\,\mathrm{sgn}(\Lambda)) \\
\dot{\hat{\Theta}}(t) &= \mathrm{Proj}(\hat{\Theta}(t), \Gamma_\Theta\, \Phi(x(t))\,\tilde{e}\,(t)PB\,\mathrm{sgn}(\Lambda))
\end{aligned} \tag{30}$$

The proceeding calculations are now summarized in Table 3.

Open-loop Plant	$\dot{x}(t) = Ax(t) + B\Lambda(u(t) + \Theta^T \Phi(x(t)))$ $y(t) = C^T x(t)$
Reference Model	$\dot{x}_m(t) = A_m x_m(t) + B_m r(t)$ $y_m(t) = C^T{}_m x_m(t)$
Tracking Error	$e(t) = x(t) - x_m(t)$
Control Signal	$u(t) = \hat{K}_x^T(t)x(t) + \hat{K}_r^T(t) - \hat{\Theta}^T(t)\Phi(x(t))$
Adaptation Laws	$\dot{\hat{K}}_x(t) = \text{Proj}(\hat{K}_x(t), -\Gamma_x x(t)\tilde{e}(t)PB\,\text{sgn}(\Lambda))$ $\dot{\hat{K}}_r(t) = \text{Proj}(\hat{K}_r(t), -\Gamma_r r(t)\tilde{e}(t)PB\,\text{sgn}(\Lambda))$ $\dot{\hat{\Theta}}(t) = \text{Proj}(\hat{\Theta}(t), \Gamma_\Theta \Phi(x(t))\tilde{e}(t)PB\,\text{sgn}(\Lambda))$
Modifications	$\tilde{e}(t) = \begin{cases} e(t) & if \quad \|e(t)\| \geq \delta \\ 0 & if \quad \|e(t)\| < \delta \end{cases}$ $\text{Proj}(\theta, y) = \begin{cases} y - \dfrac{\|\nabla f(\theta)\|^2 f(\theta) y}{} & \\ \quad \text{if}, \quad [f(\theta) > 0 \text{ and } (y\nabla f(\theta)) > 0] & \\ y \quad \text{otherwise} & \end{cases}$

Table 3. MRAC Design Summary, after [11]

B. APPLICATION: THAUS HEAVE MODE

The MRAC algorithm just developed was applied to the THAUS using the models previously estimated to represent the system dynamics. The plant model used in simulation is defined by rewriting Equation (12) in terms that better correlate with Equation (16) by adding some unmodeled parameters:

$$\dot{w} = \frac{Z_w}{m - Z_{\dot{w}}} w + \frac{Z_{w|w|}}{m - Z_{\dot{w}}} w|w| + \frac{1}{m - Z_{\dot{w}}}(Z_{prop} + g_z(\eta)) \tag{31}$$

where the linear dampening term $Z_w = 0$ is introduced to correspond to the unknown term, A. The state variable $x(t)$ corresponds to the z-direction velocity, w. System input $u(t)$ corresponds to the actuator force, Z_{prop}. It is assumed that $B = \frac{1}{m - Z_{\dot{w}}}$, which gives the nominal actuator effectiveness and is assumed to be positive, $C = 1$, and $y(t) = w(t)$

for the single input, single output system (SISO). Λ, which represents the actual actuator effectiveness, remains unknown but is assumed to be positive, and the remaining terms are written in terms of the regressor, $\Theta^T\Phi(w)$: $\Theta = [Z_{w|w|}\ g_z(\eta)]^T$ and $\Phi(w) = [w|w|\ 1]^T$.

Similarly, the reference model is specified based on the original parametric model: A_m, has been empirically evaluated and chosen to be [-0.4] in order to express the desired plant response, similar to the nominal plant; $B_m = Bk_g = \frac{k_g}{m - Z_{\dot{w}}}$; $w_m(t)$ is the model state and $r(t)$ is the command signal. Also, C=1 so that $y_m = w_m$ for the SISO system.

The control and adaptation laws maintain the same forms as those expressed in Equations (22) and (30). Both the dead zone and Projection Operator modifications are applied. The adaptation gains selected for the implementation are: $\Gamma_x = 6e6, \Gamma_r = 2e6, \text{and } \Gamma_\theta = 6e6$. The dead zone limiter is $\delta = 0.001$. The Projection Operator bounds are: $\theta_{max-x} = 100, \theta_{max-r} = 200, \text{and } \theta_{max-theta} = 50$, while the projection tolerance assigned is $\varepsilon = 0.3$. The initial conditions for the controller parameters themselves are all zero (in order to demonstrate parameter learning and convergence). Additionally, in order to better mimic the behavior of the real THAUS, a saturation limitation is placed on the output of the control signal, restricting the control effort to ± 85 Newtons.

C. SIMULATION

The MRAC system was first evaluated in simulation using MATLAB and Simulink before implementation on the THAUS. After the development of the MRAC formulae and application to the system model, a Simulink model was created to represent the closed-loop controller-plant system, shown in Figure 9. Detailed simulation models and associated code are available in Appendix A. For the real system, the plant is replaced by the commands sent to the vehicle and the actual measurements. This process and associated modifications are described in Chapter V.

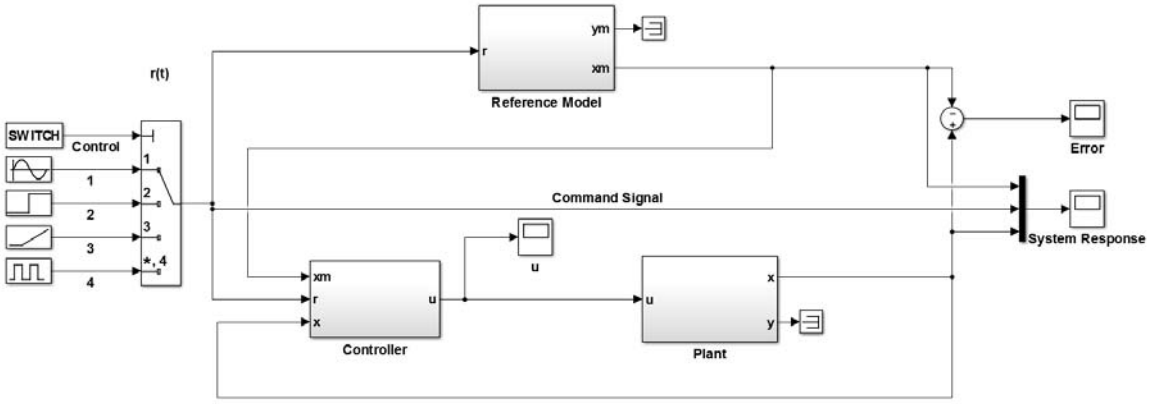

Figure 9. MRAC Simulink Model

1. MRAC: Sinusoidal Command

The first simulation evaluates the MRAC system's response to the sinusoidal command, $r(t) = \frac{1}{2}\sin(0.1t + \frac{\pi}{2})$. The simulation results in Figure 10. show a plant response the follows the reference model fairly well, without significant error. Due to the choice of the parameters in the reference model, the reference model lags the command signal; however, the actual plant follows the reference model well. The initial deviation occurs during the parameter learning phase wherein the adaptive system learns the unknown plant model.

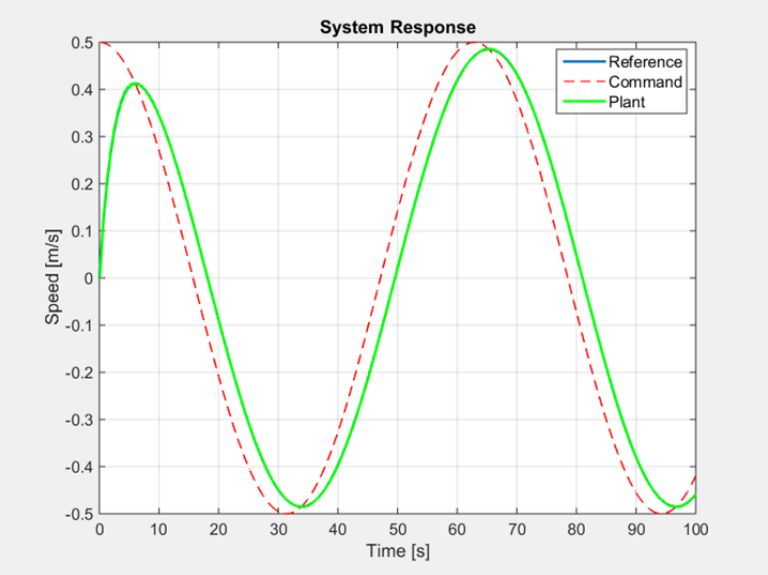

Figure 10. MRAC: Sinusoidal Command System Response

The error, $e(t)$, present in the system response, shown in Figure 11. diminishes quickly and then remains reasonably small. Error is highest during the parameter learning phase, as expected. At steady state error lies almost entirely within the permissible dead zone.

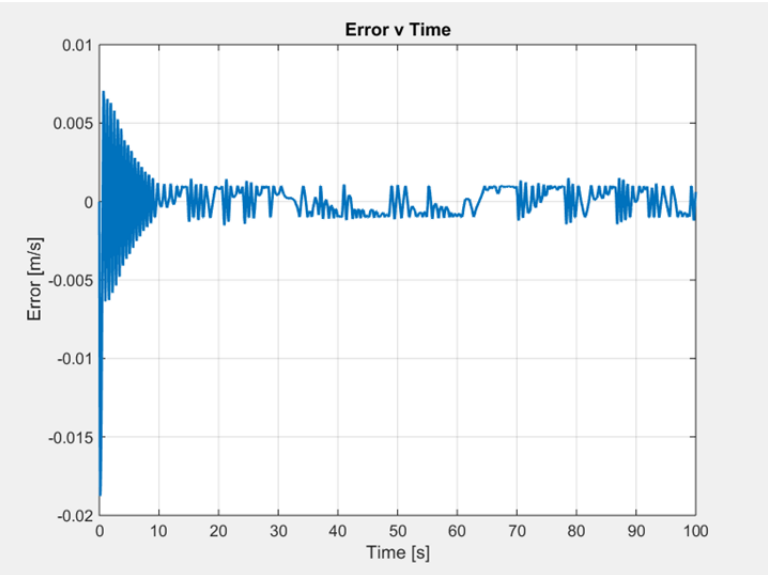

Figure 11. MRAC: Sinusoidal Command Error

The controller parameters in Figure 12. did not converge to steady state values in the time frame of this simulation, but rather oscillate around their solutions.

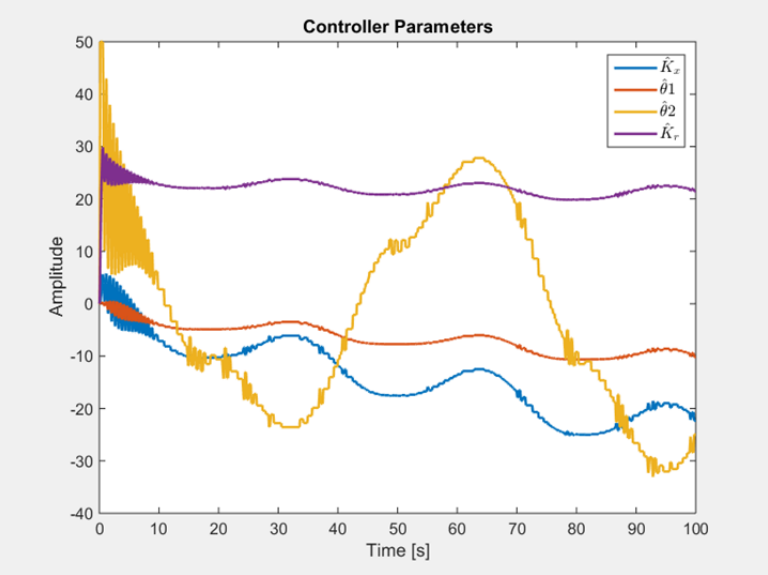

Figure 12. MRAC: Sinusoidal Command Controller Parameters

The control signal is shown in Figure 13. and varies according to the reference model behavior. Initially, the control signal oscillates greatly, before controller parameters converge around their solutions. In steady state, the response remains relatively smooth overall but contains undesirable, low amplitude, high frequency content and, importantly, is well bounded within the abilities of the system.

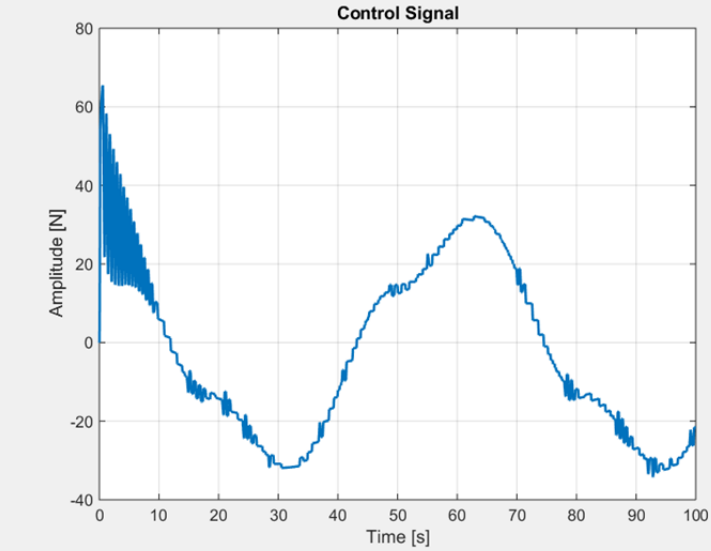

Figure 13. MRAC: Sinusoidal Command Control Signal

The simulated response of the MRAC system demonstrates the control system's ability to follow a reference command with limited a priori knowledge of the plant's dynamic parameters.

2. MRAC: Time-Varying Parameters

The second simulation evaluates the MRAC system's ability to respond to a step response with $r = 0.5$ m/s and time-varying parameters based on the addition of a 0.02 kg mass at $t = 15$ s. The system responds well both to the step command as well as the mass change. The parameters of the reference model remain constant; therefore the response of the reference model is unaffected and remains steady. Only plant parameters are changed in order to demonstrate the controller's adaptability.

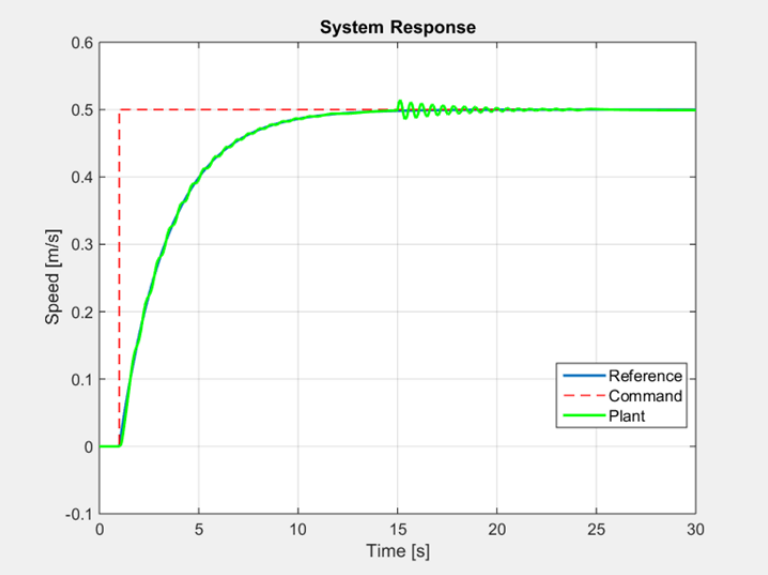

Figure 14. MRAC: Time-Varying Parameters System Response

Though oscillations about the reference model exist, the system reaches a steady state error within the dead zone's δ, as seen in Figure 15.

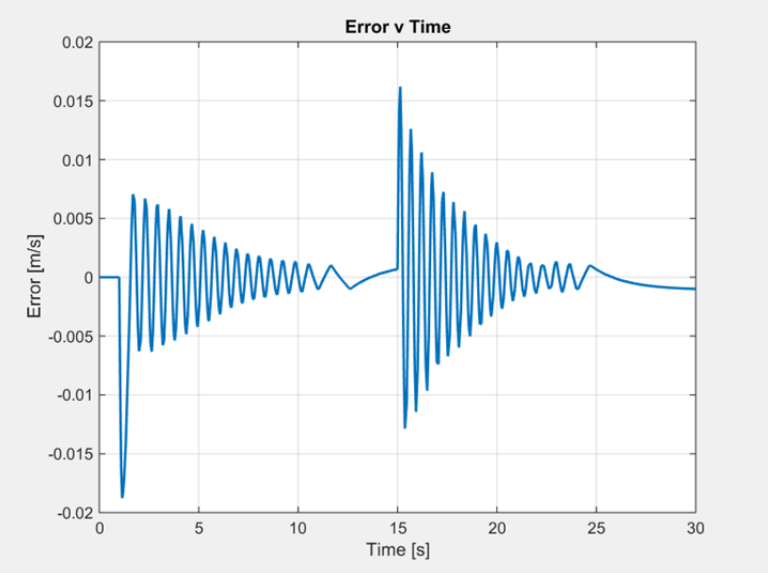

Figure 15. MRAC: Time-Varying Parameters Error

In this instance of simulation, the adaptation law converges on a controller parameter solution; however, these parameters are adjusted in response to the

configuration change and settle at new solutions. Controller parameters are shown in Figure 16. The change in plant parameters themselves can be seen in Figure 17.

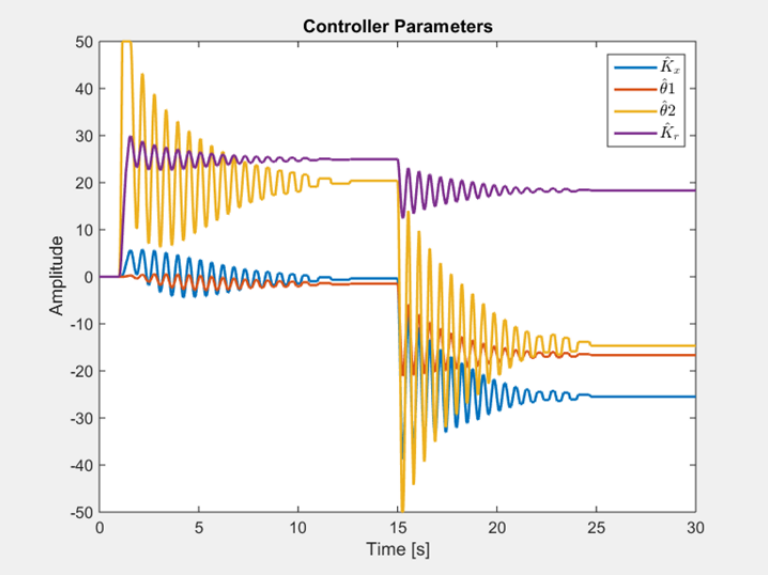

Figure 16. MRAC: Time-Varying Parameters Controller Parameters

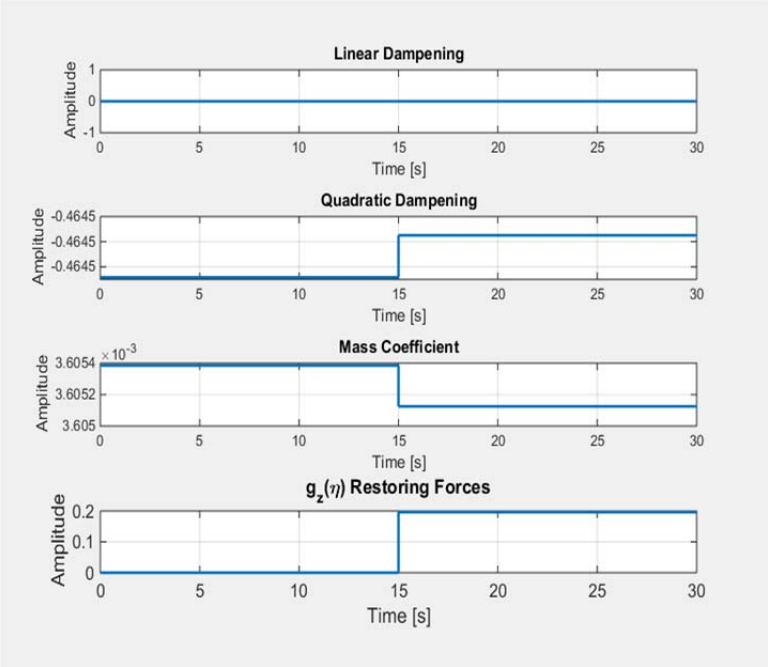

Figure 17. MRAC: Time-Varying Parameters

The control signal, shown in Figure 18. , compensates as expected to the step response and reaches a steady state value as the vehicle's velocity converges to the command. Upon changing the mass, the control signal changes to compensate for the additional force of the increased mass.

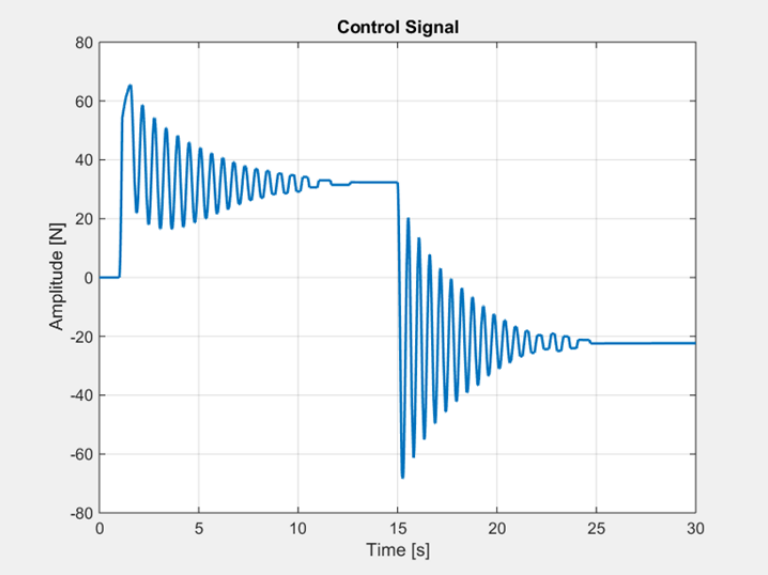

Figure 18. MRAC: Time-Varying Parameters Control Signal

Again, the MRAC system implementation demonstrates the control system's ability to respond to command signals as well as a plant configuration change. Furthermore, tuning of the control gains, and bounds will improve the system response; yet, limitations on the system's potential do exist, as high frequency content is again present.

D. LIMITATIONS

While the MRAC system has the potential to effectively control a THAUS in the presence of uncertain, time-varying parameters and disturbances, its architecture is not without drawbacks. The most prominent limitation is the direct coupling of robustness and adaptation rate (i.e., the speed with which it can react to changes in the system). Fast adaptation and response requires high gains that drive the control system to respond quicker, but with a stronger control effort. A robust response requires low gains, allowing

the controller to more gradually adapt and arrive at convergence without overshoot and with minimal control effort. Because these two mechanisms are coupled, a balance between them must be met, and performance will be affected. Optimal performance is achieved by tuning the controller parameters. If a system is tuned to favor speed, robustness will be sacrificed. The control response will be fast but will likely overshoot its target, resulting in high frequency, oscillatory behavior prior to convergence. In fact, MRAC does not provide any guarantees on the transient response of the system. If a system is tuned to favor robustness, speed will be sacrificed. The control response will be smooth, and restrained, but may struggle to keep up with varying parameters or changing commands. Achieving steady state tracking will also take longer since the control effort is unaggressive. It is difficult to find an optimal solution in the tuning process since no proven algorithms exist for that purpose. Control parameters are often selected through empirical evaluation or engineering intuition [25], [26].

The stability of the system response is normally related to the parameter estimation process. Parameter estimate drift can cause the estimates to become unbounded. The addition of modifiers to the adaptation law, as shown in Section A of this chapter, alleviates some of the limitations of the speed/robustness coupling. The dead zone modification and the Projection Operator modification both allow for higher gains to be implemented by constraining the estimates to remain within a bounded set.

Each modifier accomplishes this goal in its own way, but both require sacrifices to steady state tracking error. The dead zone modification establishes an acceptable value of error thus leading to potentially constant steady state errors that the system ignores [10]. Similarly, the Projection Operator has been shown to increase robustness when augmented onto an MRAC system by bounding the control parameters; however, bounding the parameters prevents the system from establishing a true solution. A solution vector ending outside the projection bounds is always redirected along the edge of the boundaries. This results in parameters that never truly solve the equation in (20) because the estimate error may never converge to zero, $|\Delta\Theta| > 0$ [11]. Thus, the cost for keeping the estimates bounded is at least a temporary deviation from the optimal reduction in output error.

As a potential solution to these limitations, $\mathcal{L}_1$ AC is evaluated in the next chapter. This alternative approach effectively decouples the adaptation properties of the controller from the stability of the controller, allowing for fast adaptation. Furthermore, the approach establishes some bounds on the transient response of the system.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. $\mathcal{L}_1$ ADAPTIVE CONTROL SYSTEM

A. DEVELOPMENT

1. Overview

Originally proposed by Hovakimyan and Cao, the $\mathcal{L}_1$ AC architecture decouples adaptation rate from robustness while obtaining guaranteed transient response. Additionally, $\mathcal{L}_1$ AC eliminates the need for gain scheduling, persistence of excitation or high-gain feedback control that may result in high frequency oscillations in the control signal. In other words, $\mathcal{L}_1$ AC attempts to address many of the limitations associated with MRAC.

Like MRAC, an $\mathcal{L}_1$ AC system consists of four main components. The *plant* is the target system to be controlled, containing unknown or unmodeled dynamic parameters. It is again useful to distinguish between matched and unmatched uncertainty in the system: this work deals only with matched uncertainty. The basic structure of the plant's dynamic model is generally known (normally derived from first principle). The *predictor* is similar to the MRAC reference model, in that it represents the ideal, achievable system behavior that is to be imitated by the efforts of the adaptive system. One important difference between the reference model and the predictor system is that the generated control signal enters both the predictor and the plant, allowing additional flexibility in the design of the control law (as compared to MRAC) and, importantly, the transient response of the system. The predictor's output is compared to the plant for use during adaptation (i.e., the *adaptation law*). The adaptation law, composed of a Projection Operator based equation (as in the MRAC modification), takes input from the plant-predictor error and adjusts the control parameters, eventually reaching convergence in a system with constant parameters. The *control law* generates a control signal, based on commanded input and a signal composed of the adaptive gains (defined explicitly below). This control signal is passed through a low pass filter, to eliminate high frequency content and increase robustness. Again, as control parameters converge to a solution that satisfies the dynamic

equation, tracking error will trend toward zero [25], [27]. The general architecture for an $\mathcal{L}_1$ AC system is shown in Figure 19.

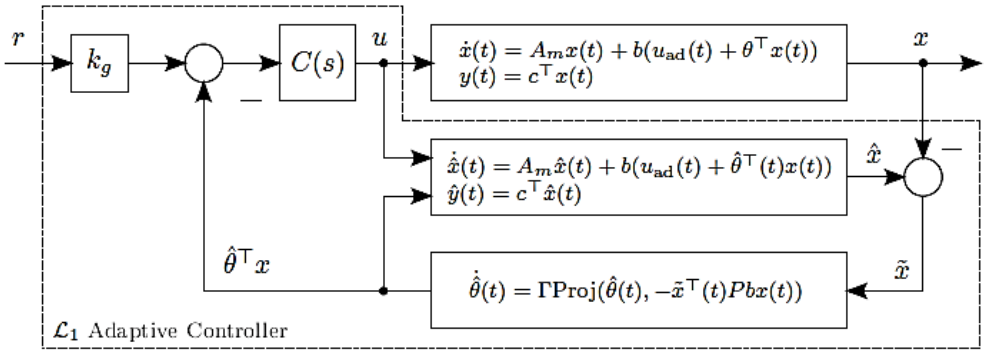

Figure 19. $\mathcal{L}_1$ Adaptive Control General Architecture, from [25]

2. System Plant

The $\mathcal{L}_1$ AC plant takes the general form,

$$\begin{aligned} \dot{x} &= A_m x + B(\omega u(t) + f(t, x(t))) \qquad x(0) = x_0 \\ y &= C^T x(t) \end{aligned} \tag{32}$$

where $x(t) \in \mathbb{R}^n$ is the system's measured state; $A_m \in \mathbb{R}^{n \times n}$ is a known Hurwitz matrix expressing the desired closed-loop dynamics; B and $C \in \mathbb{R}^n$ are known constant vectors; $u(t) \in \mathbb{R}$ is the control signal; $\omega \in \mathbb{R}$ and $\in \Omega \triangleq [\omega_{min}, \omega_{max}]$ is an unknown parameter representing uncertain command efficiency and are known bounds, such that $0 < \omega_l < \omega_u$; $f(t, x(t)) : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is an unknown, nonlinear, continuous mapping; $x_0 \in \mathbb{R}$ is the state initial condition, known to be in an arbitrarily large, but bounded set; and $y(t) \in \mathbb{R}$ is the system output [25].

3. State Predictor

The state predictor builds off of the basic plant architecture and represents the estimated states:

$$\begin{aligned} \dot{\hat{x}} &= A_m \hat{x}(t) + B(\hat{\omega}(t)u(t) + \hat{\theta} \| x(t) \|_\infty + \hat{\sigma}(t)) \qquad x(0) = x_0 \\ \hat{y} &= C^T \hat{x}(t) \qquad\qquad\qquad\qquad\qquad\qquad\qquad | \end{aligned} \tag{33}$$

where $\hat{\omega}(t), \hat{\theta}(t), \text{and} \, \hat{\sigma}(t) \in \mathbb{R}$ are the adaptive parameter estimates [25]. Here the infinity norm of the state variable, $\| x(t) \|_\infty$ is a suitable substitute for the nonlinear term, $x(t) | x(t) |$, as shown in Section 2.4 in [25].

4. Control Law

The $\mathcal{L}_1$ AC system composes its control signal according to the following Laplace domain Equation [25]:

$$\begin{aligned} u(s) &= -kD(s)(\hat{\mu}(s) - k_g r(s)) \\ \hat{\mu}(s) &= \mathcal{L}\{\hat{\mu}\} \triangleq \mathcal{L}\{\hat{\omega}(t)u(t) + \hat{\theta} \| x(t) \|_\infty + \hat{\sigma}(t)\} \end{aligned} \tag{34}$$

where $D(s)$ is a strictly proper transfer function acting as a low pass filter, and k is the feedback gain; the selection criteria for $D(s)$ and k are discussed further in Section 6. The command signal gain, $k_g = \dfrac{-1}{C^T A_m^{-1} B}$, satisfies zero steady state tracking error for a step response.

5. Adaptation Law

The control law above is written in terms of the estimated parameters. Calculation of the adaptive parameter estimates is accomplished according to the laws shown in Equation (35). The Projection Operator in the adaptation law ensures that the adaptive parameter estimates remain bounded, such that $\hat{\omega}(t) \in \Omega, | \hat{\theta}(t) | \leq \theta_{max}$ and $| \hat{\sigma}(t) | \leq \sigma_{max}$. This is required for the stability proof to remain valid.

$$\begin{aligned} \dot{\hat{\theta}}(t) &= \Gamma_\theta \text{Proj}(\hat{\theta}(t), -\tilde{x}^T(t) \| x(t) \|_\infty PB) \qquad \hat{\theta}(0) = \hat{\theta}_0 \\ \dot{\hat{\sigma}}(t) &= \Gamma_\sigma \text{Proj}(\hat{\sigma}(t), -\tilde{x}^T(t) PB) \qquad \hat{\sigma}(0) = \hat{\sigma}_0 \\ \dot{\hat{\omega}} &= \Gamma_\omega \text{Proj}(\hat{\omega}(t), -\tilde{x}^T(t) u(t) PB) \qquad \hat{\omega}(0) = \hat{\omega}_0 \end{aligned} \tag{35}$$

where, $\Gamma \in \mathbb{R}^+$ is the adaptation gain. State error, $\tilde{x}(t)$, is defined by

$$\tilde{x}(t) \triangleq \hat{x}(t) - x(t) \tag{36}$$

and $P = P^T > 0$ is the solution to the Lyapunov equation, $A_m^T P + P A_m = -Q$, with arbitrary, symmetric $Q = Q^T > 0$ [25].

The proceeding calculations are now summarized in Table 4.

Open-loop Plant	$\dot{x} = A_m x + B(\omega u(t) + f(t, x(t)))$ $\quad x(0) = x_0$ $y = C^T x(t)$	
State Predictor	$\dot{\hat{x}} = A_m \hat{x}(t) + B(\hat{\omega}(t)u(t) + \hat{\theta} \Vert x(t) \Vert_\infty + \hat{\sigma}(t))$ $\hat{y} = C^T \hat{x}(t)$	$x(0) = x_0$
Tracking Error	$\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$	
Control Signal	$u(s) = -kD(s)(\hat{\mu}(s) - k_g r(s))$ $\hat{\mu}(s) = \mathscr{L}\{\hat{\mu}\} \triangleq \mathscr{L}\{\hat{\omega}(t)u(t) + \hat{\theta} \Vert x(t) \Vert_\infty + \hat{\sigma}(t)\}$	
Adaptation Laws	$\dot{\hat{\theta}}(t) = \Gamma_\theta \text{Proj}(\hat{\theta}(t), -\tilde{x}^T(t) \Vert x(t) \Vert_\infty PB)$ $\dot{\hat{\sigma}}(t) = \Gamma_\sigma \text{Proj}(\hat{\sigma}(t), -\tilde{x}^T(t) PB)$ $\dot{\hat{\omega}} = \Gamma_\omega \text{Proj}(\hat{\omega}(t), -\tilde{x}^T(t) u(t) PB)$	$\hat{\theta}(0) = \hat{\theta}_0$ $\hat{\sigma}(0) = \hat{\sigma}_0$ $\hat{\omega}(0) = \hat{\omega}_0$

Table 4. $\mathcal{L}_1$ AC Design Summary, after [11]

B. APPLICATION: THAUS HEAVE MODE

The parametric model from Equation (12) is rewritten in the form of the $\mathcal{L}_1$ AC plant architecture from Equation (32):

$$\dot{w} = \frac{Z_w}{m - Z_w} w + \frac{Z_w |w|}{m - Z_w} w |w| + \frac{1}{m - Z_w} (Z_{prop} + g_z(\eta)) \tag{37}$$

where, the linear damping term, $Z_w = 0$ is introduced to correspond to A_m, which has been empirically evaluated to be [-0.4] in order to express the actual (measured) plant response. The state variable $x(t)$ corresponds to the z-direction velocity, w. System input $u(t)$ corresponds to the actuator force, Z_{prop}. It is conveniently assumed that $B = \frac{1}{m - Z_{\dot{w}}}$, also $C = 1$ for the single input, single output system (SISO). The unknown constant, ω, represents the actuator effectiveness, an uncertain system input gain. The remaining

terms from the parametric model can be associated with the unknown function, $f(t, x(t))$:

$$f(t, x(\mathrm{t})) = (\frac{Z_w}{m - Z_{\dot{w}}} - A_m) w(t) + \frac{Z_{w|w|}}{m - Z_{\dot{w}}} w(t) \mid w(t) \mid + \frac{1}{m - Z_{\dot{w}}} g_z(\eta) \tag{38}$$

Combining and rewriting these terms in the form of Equation (32) results in the plant dynamics expression

$$\dot{w}(t) = A_m w(t) + B(\frac{1}{m - Z_{\dot{w}}} u(t) + (\frac{Z_w}{m - Z_{\dot{w}}} - A_m) w(t) + \frac{Z_{w|w|}}{m - Z_{\dot{w}}} w(t) \mid w(t) \mid + \frac{1}{m - Z_{\dot{w}}} g_z(\eta)), \qquad w(0) = w_0 \tag{39}$$
$$y(t) = w(t)$$

which simplifies to the original parametric model for the THAUS from Equation (12).

Similar to the plant, the predictor term $A_m = [-0.4]$ is chosen to give the desired plant response, $B = [\frac{1}{m - Z_{\dot{w}}}]$ is assumed, $C = [1]$ and $\hat{y}(t) = \hat{x}(t)$ for a SISO system. The term, $\| x(t) \|_\infty$, is the infinity norm of the states and corresponds to $\| w(t) \|_\infty = | w(t) |$ for the SISO system and replaces the nonlinear term. The remaining terms in the predictor do not correlate directly with terms in the parametric model and are adjusted by the adaptive and control laws. The predictor equation can now be rewritten as:

$$\begin{aligned} \hat{\dot{w}}(t) &= \mathrm{A}_m w(t) + \mathrm{B}(\hat{\omega}(t) u(t) + \hat{\theta}(t) \mid w(t) \mid + \hat{\sigma}(t)) \qquad w(0) = w_0 \\ \hat{y}(t) &= \hat{w}(t) \end{aligned} \tag{40}$$

No significant changes to the adaptation or control laws, other than substituting the correct states, were required. The adaptation gain applied was $\Gamma = 5e7$. The Projection Operator bounds are $\theta_{max-\sigma} = 400, \theta_{max-\theta} = 200, \text{and } \theta_{max-\omega} = 0.85$, with ω parameter Projection Operator centered at 1.15. The initial conditions for the controller parameters are all zero. The control signal filter is $D(s) = \frac{1}{s}$, and the feedback gain, $k = 6$. Furthermore, in order to better mimic the behavior of the real THAUS, a

saturation limitation is placed on the output of the control signal, restricting the control effort to ± 85 Newtons.

C. SIMULATION

The $\mathcal{L}_1$ AC system was evaluated in simulation using MATLAB and Simulink, without implementation on the THAUS. After the development of the $\mathcal{L}_1$ AC algorithm and application to the system model a Simulink model was created to represent the closed-loop controller-plant system, shown in Figure 20. Detailed simulations models and associated code are available in Appendix B.

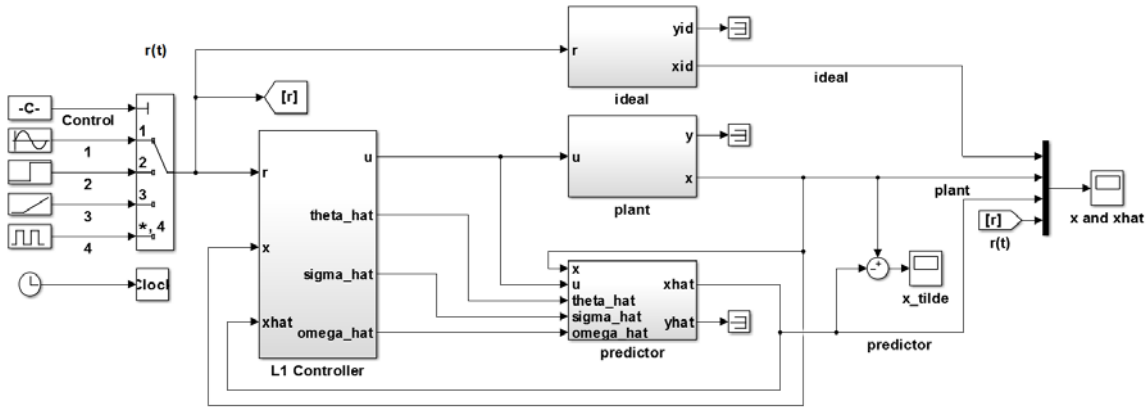

Figure 20. $\mathcal{L}_1$ AC Simulink Model

1. $\mathcal{L}_1$ AC: Sinusoidal Command

The first simulation evaluates the $\mathcal{L}_1$ AC system's response to the sinusoidal command, $r(t) = \frac{1}{2}\sin(0.1t + \frac{\pi}{2})$. The simulation results in Figure 21. show a plant response the follows the state predictor extremely closely, without significant error. The deviation from the command signal is, similar to MRAC, primarily a result of the predictor model's limitations in its ability to respond to and follow the command signal. Comparison to the ideal output confirms this.

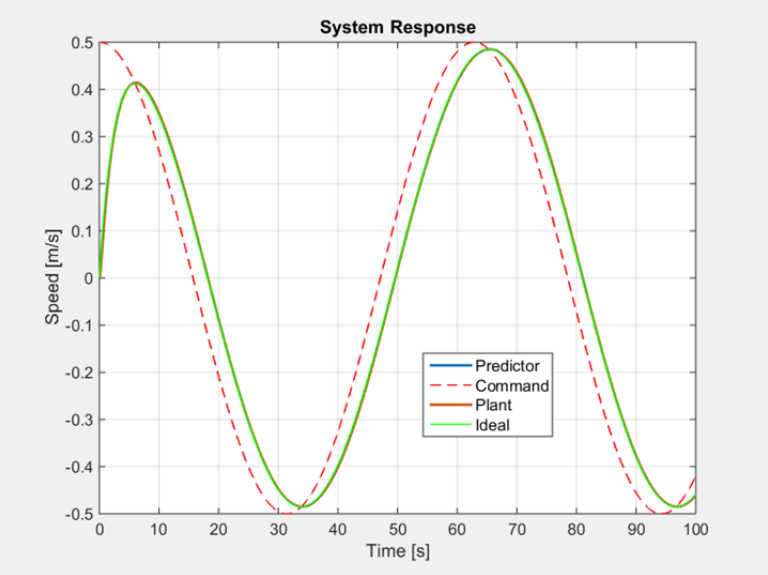

Figure 21. $\mathcal{L}_1$ AC: Sinusoidal Command System Response

The error data (Figure 22.) reveals some small oscillations of the plant response about the predictor's response; however, this error is much smaller than THAUS' sensors are cable of measuring and is considered insignificant.

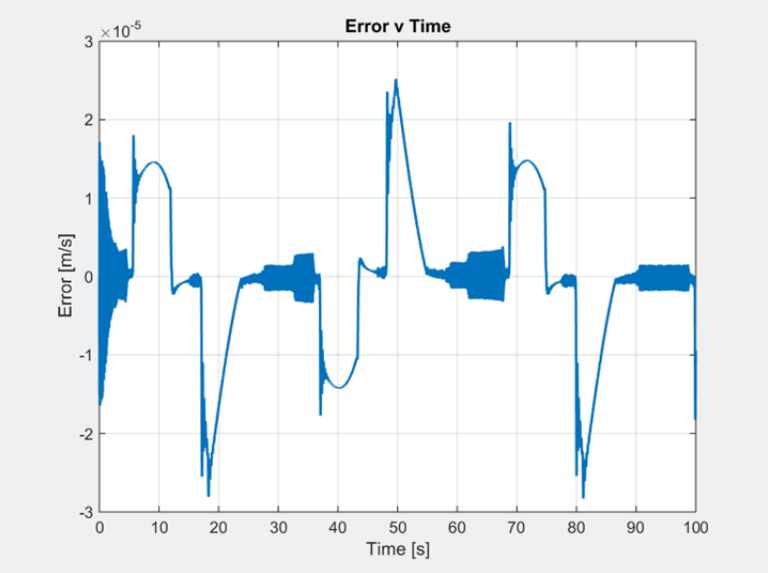

Figure 22. $\mathcal{L}_1$ AC: Sinusoidal Command Error

Although controller parameters in Figure 23. do not converge to a steady state solution within the simulation time frame they do appear to oscillate, with plateaus, about solution value; this boundedness permits steady state error to go to zero. The various values for the controller parameters each satisfy the adaptation solution, achieving very small tracking error.

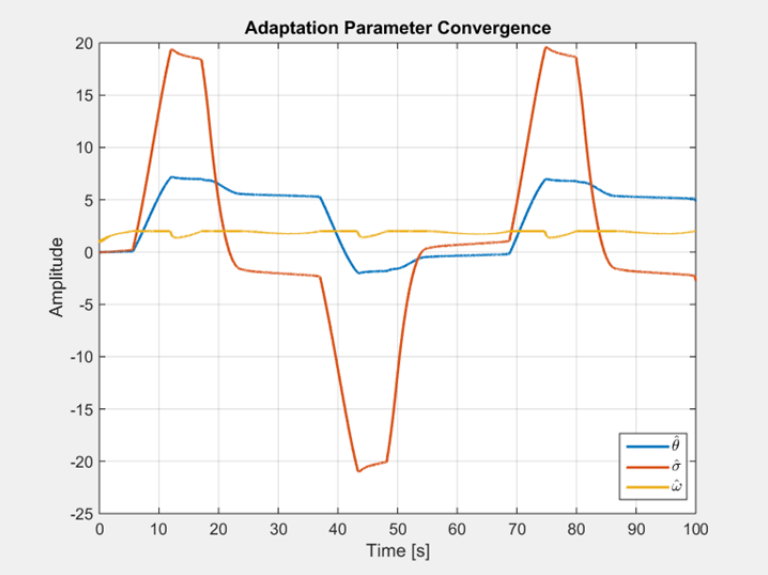

Figure 23. $\mathcal{L}_1$ AC: Sinusoidal Command Controller Parameters

The control signal in Figure 24. remains smooth and bounded throughout the simulation. The control effort required by the system is achievable.

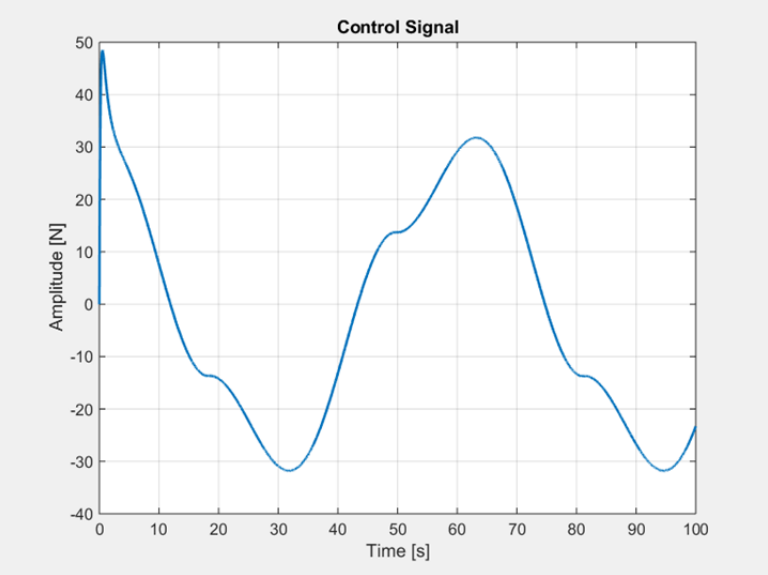

Figure 24. $\mathcal{L}_1$ AC: Sinusoidal Command Control Signal

This simulation demonstrates the successful response of the $\mathcal{L}_1$ AC system when applied to a system with limited a priori knowledge of the plant's dynamic parameters.

2. $\mathcal{L}_1$ AC: Time-Varying Parameters

The second simulation evaluates the $\mathcal{L}_1$ AC system's ability to respond to a step response with $r = 0.5$ m/s, and time-varying parameters based on the addition of a 0.02 kg mass at $t = 15$ s. The system responds well both to the step command as well as the mass change, shown in Figure 25. Because the controller responds quickly the effect of the configuration change is relatively small. In comparison with the MRAC system, the transient response is much more stable, while also very quick to respond and reach steady state. Additionally, while the MRAC reference model does not respond to the change in parameters (as discussed in Chapter III) the predictor system does. This response is due to the differing structure of the $\mathcal{L}_1$ adaptive controller. The control signal, $u(t)$, a function of the parameter estimates, enters both the predictor as well as the plant. A change in actual parameters results in a change in their estimates and thus a change in predictor behavior.

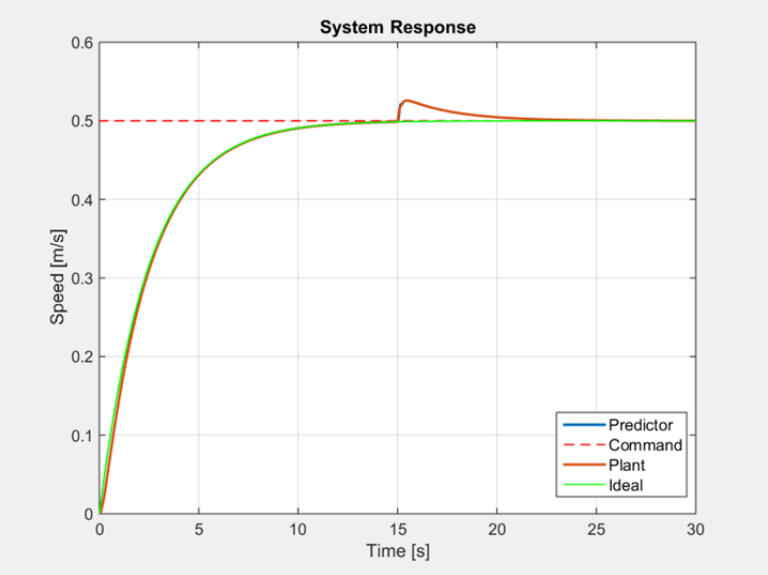

Figure 25. $\mathcal{L}_1$ AC: Time-Varying Parameters System Response

Figure 26. shows the error to be very small with only slight divergences from state predictor's response. The only significant error is that induced by the sudden parameter change, though this is quickly corrected by the controller.

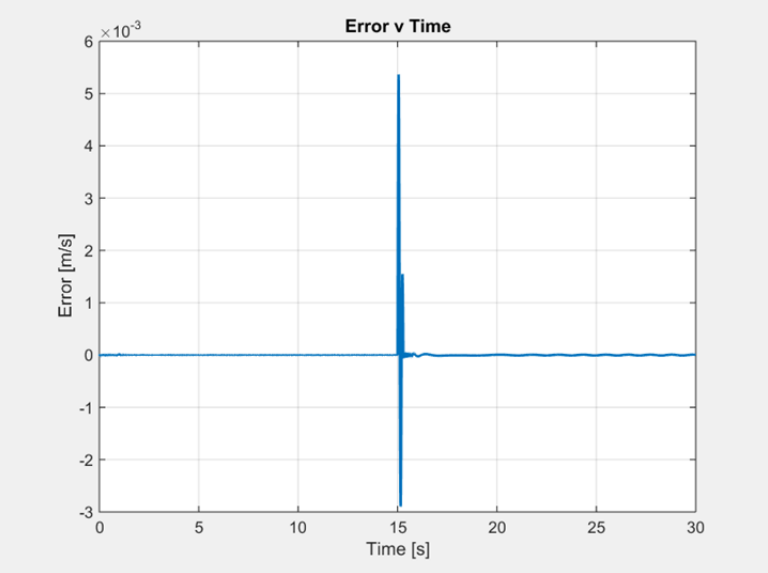

Figure 26. $\mathcal{L}_1$ AC: Time-Varying Parameters Error

In Figure 27. the controller parameters quickly converge to steady state solutions. Once the mass change is implemented the parameters adjust and quickly converge to new state solutions. Figure 28. shows the dynamic parameters themselves changing.

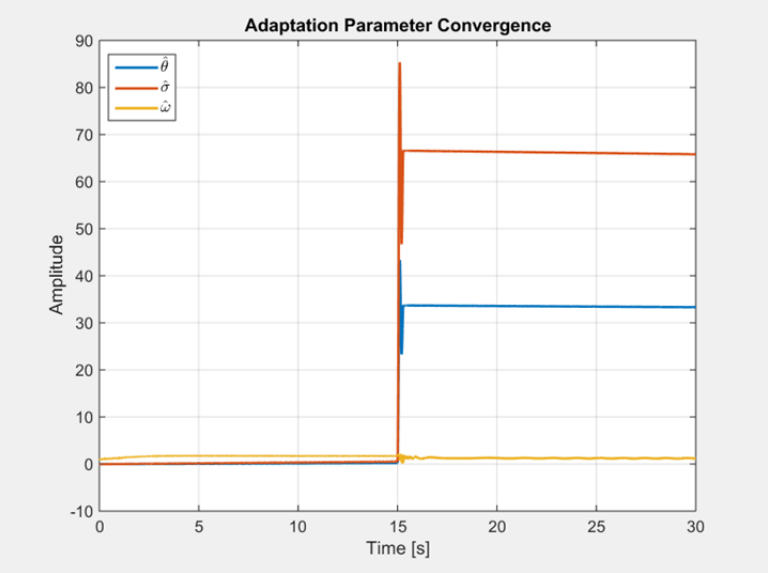

Figure 27. $\mathcal{L}_1$ AC: Time-Varying Parameters Controller Parameters

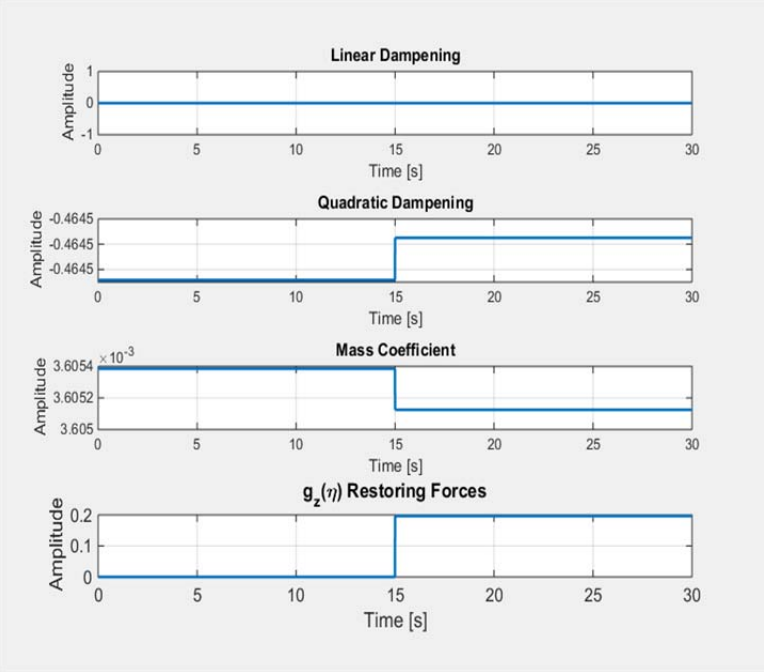

Figure 28. $\mathcal{L}_1$ AC: Time-Varying Parameters

The control signal, shown in Figure 29. , applied throughout simulation is bounded, maintaining an achievable control effort. After the mass is increased, the system responds quickly and settles at a new steady state control value. Some very small amplitude, undesirable, high frequency content was observed in the control signal; however, this never translated into high frequency responses by the plant or state predictor. More careful selection of the control signal filter could potentially eliminate this behavior.

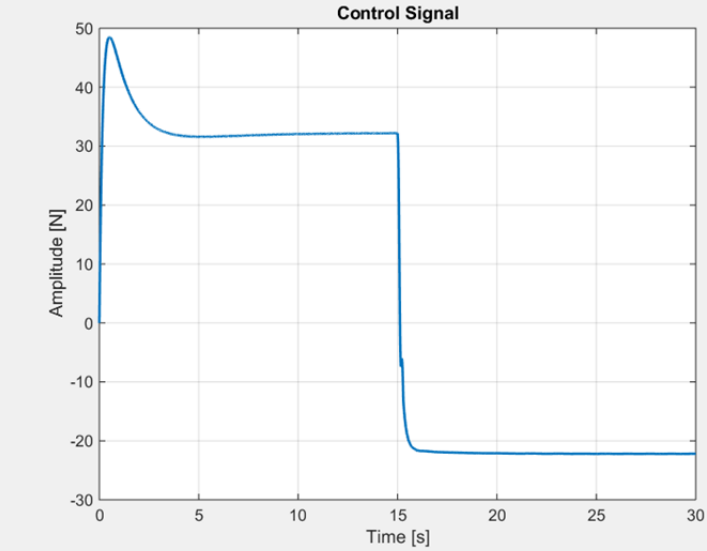

Figure 29. $\mathcal{L}_1$ AC: Time-Varying Parameters Control Signal

The $\mathcal{L}_1$ AC system simulation with time-varying parameters successfully demonstrated the control system's ability to respond to step command signals as well as a plant configuration change. The high gains employed in this controller enabled the system to track the state predictor quickly and accurately, using an achievable control effort. In comparison to MRAC, transient responses were much quicker, with fewer oscillations and faster settling times.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EXPERIMENTAL RESULTS

A. IMPLEMENTATION

Testing the developed algorithms on the THAUS platform requires integration with the command software used for the autonomous operation of the platform. This infrastructure has been developed as part of the RDAS research program. The Robotic Operating System (ROS) is used for this purpose. ROS is foremost a development and deployment architecture that provides a messaging protocol to communicate information between processes. Since ROS uses a TCP/IP protocol, these processes can be distributed across multiple machines on a network. ROS additionally provides tools for the development and execution of these systems, which consist of multiple processes. This architecture allows for the development of highly modular code to realize a complex robotic system. Researchers at CAVR have implemented such architecture (Figure 30.) on the THAUS. One of the benefits of this modular implementation is the ability to apply the same tools on multiple platforms, including the REMUS AUV, SeaFox USV, and the ScanEagle UAV.

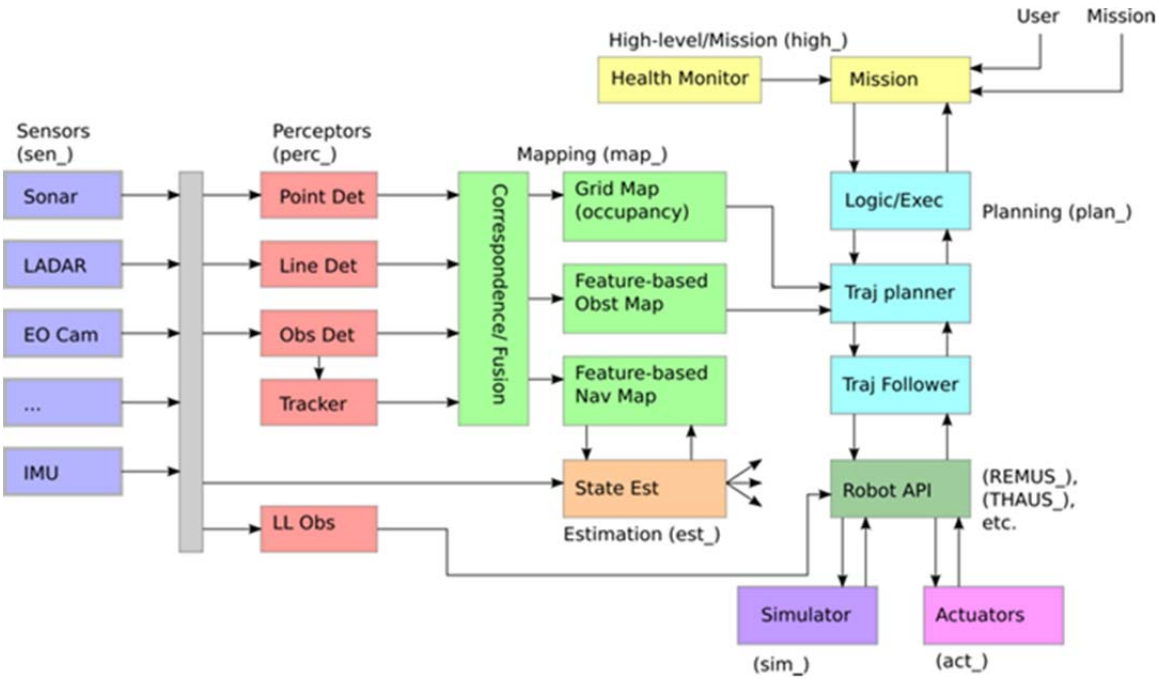

Figure 30. ROS Software Architecture

One prohibitive factor with ROS is the requirement to develop in either the C++ or Python programming languages, which significantly increases the learning curve to be able to implement advanced control algorithms as part of student research. Instead, one convenient tool that is routinely used at NPS for control algorithm development is MATLAB Simulink. This provides a useful graphical interface for designing and implementing algorithms. In parallel to the efforts of this research, CAVR personnel have developed an approach to directly integrate the algorithms developed in Simulink into the ROS architecture. The Simulink model for the ROS interface is shown in Figure 31. This approach utilizes the automatic code generation functionality of Simulink to build a C++ library containing the algorithm. This library is then integrated into a process (ROS node) that provides the control algorithm with the necessary information (i.e., measurements, reference signals, and controller gains). This node, implemented in C++, initializes the control algorithm and schedules its execution.

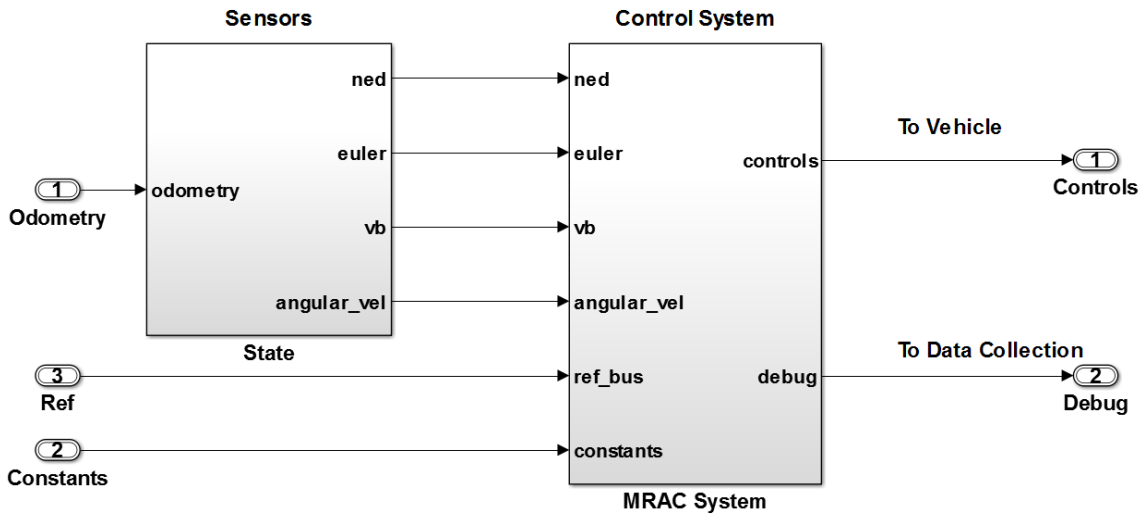

Figure 31. ROS Interface Simulink Model

Data exchange is defined through the use of *buses* in Simulink. In the implementation above, three input buses were defined: odometry (measured vehicle states), reference signal, constants (for controller tuning). Two output buses include the control signal and generic debug information. Initial conditions and other control algorithm parameters can be set from within the ROS process as well.

The automatic code generation functionality was developed to support real-time execution of control processes. However, most stock operating systems do not support *hard real-time* execution by default. This limitation has become less severe as operating systems have improved and processor speeds have increased. For Linux, these algorithms can be run in *soft real-time*, which most often proves sufficient. The current implementation of this ROS-Simulink integration does not provide hard real-time execution. This will be investigated in the future if the need arises.

Soft real-time execution requires careful consideration of the sampling times used when developing the control algorithm. For this research, the system amounts to two loops that run at different rates. For the adaptation process, there is a need to increase the execution rate as much as possible to allow for fast adaptation. However, this process must interface with a real platform that can only execute commands at a specified (lower) rate, and for which measurements can only be obtained at a much slower rate.

Furthermore, the real system will have delays due to communications, computation, actuator dynamics, etc. These differences between the simulated system and the real system must be accounted for as much as possible and is investigated in Section B.1.

For this implementation, a fixed-step numerical integration scheme (the ode3 algorithm) was used. This time step was set to 0.01 s sampling time. This sampling time is limited by the amount of computation required to calculate the adaptation and control signal for the high-rate loop. This sampling rate was verified to be sufficient, but was not optimized. This task will be undertaken in future work. The ROS node schedules the inner-loop control algorithm and communicates the control signal to the vehicle at the appropriate times.

B. EXPERIMENTAL RESULTS

1. MRAC: Time-Varying Parameters Simulation with Rate Transitions

An additional step was taken in the MRAC simulation model that added rate transitions of signals between the plant, reference model and controller blocks. The rate transitions were added in order to better simulate the actual processing ability of the THAUS. The system is able to send commands at 10 Hz, and can receive data at 12 Hz. In this simulation, the gains were $\Gamma_x = 1e5, \Gamma_r = 1e5, \text{and } \Gamma_\theta = 1e5$. No other conditions were changed. Figure 32. shows the response of the system to a step response with time-varying parameters, but without rate transitions. In contrast, Figure 33. shows the system response after rate transitions have been added (to better represent the actual plant). The response with rate transitions is more oscillatory due to more aggressive corrective behavior that occurs as a result of the slower signal rates. The oscillations present in the system response are consistent in the error, parameter convergence, and error data as well. This behavior can be minimized, though, by tuning the system in such a way that limits the aggressive corrections. The reduced gains used in the final MRAC simulations (Figure 34.), $\Gamma_x = 6e4, \Gamma_r = 2e4$, and $\Gamma_\theta = 2e4$, were selected for this purpose. Applying rate transitions to simulations using the reduced gains produced a system response very similar to those with continuous signal rates. For this reason, these gains were employed in the experimental implementation discussed in Sections 2 through 4.

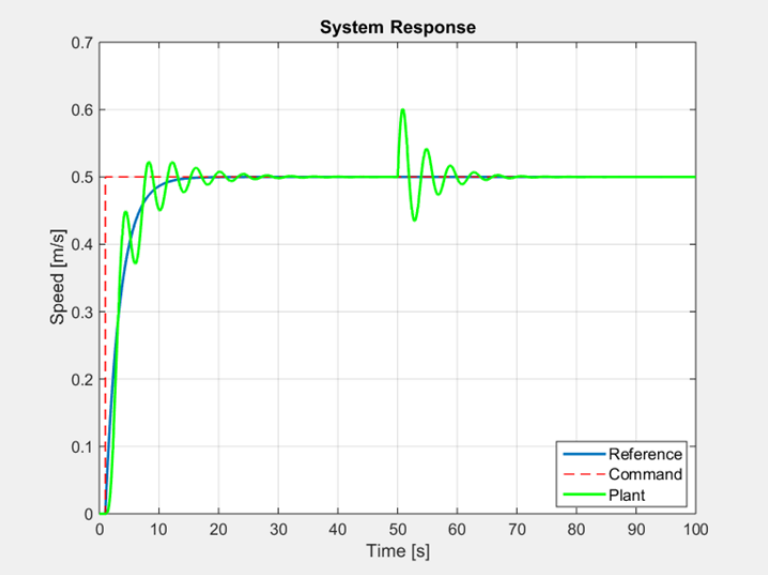

Figure 32. MRAC: Time-Varying Parameters, No Rate Transitions and Original Gains

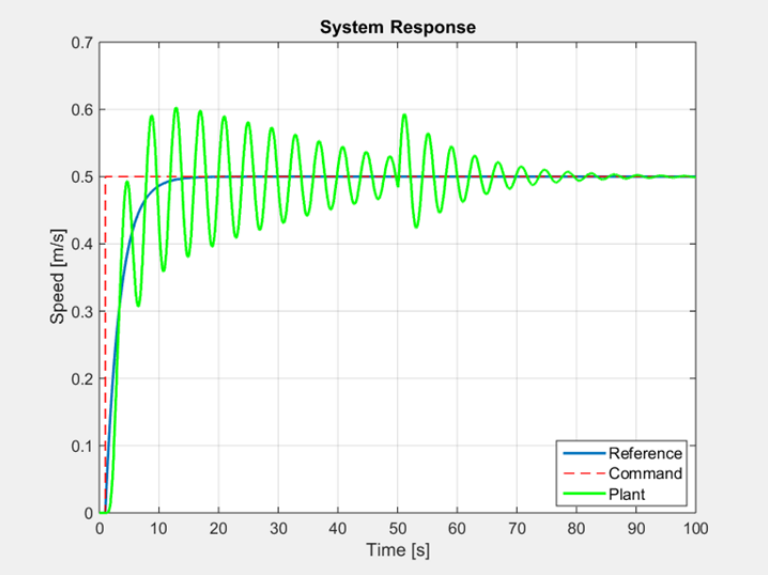

Figure 33. MRAC: Time-Varying Parameters, Rate Transitions and Original Gains

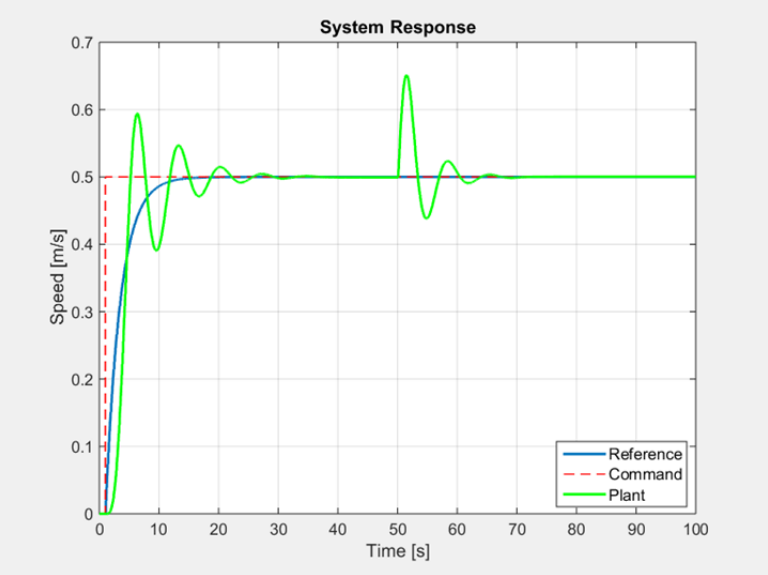

Figure 34. MRAC: Time-Varying Parameters, Rate Transitions and Tuned Gains

2. MRAC Depth Step Command

The physical experimentation phase of research began with the MRAC system implemented on board the vLBV300 THAUS via the ROS interface as discussed in Section A. The experiments took place in a small tank of fresh water, approximately 1.5 m deep. First, the vehicle was given a depth step command, moving from $z = 0.1$ m to $z = 0.4$ m depth. This position command was translated into a rate command by the proportional relationship, $w_{command} = 0.2 \times z_{error}$. The gains used to implement the controller were the same as those developed in simulation, $\Gamma_x = 6e4, \Gamma_r = 2e4$, and $\Gamma_\theta = 2e4$. The system response is shown in Figure 35. . As expected, oscillations are present in both the command velocity and the plant velocity. While the plant was tuned to maximize robustness, not all oscillatory behavior could be eliminated. Additionally, noise and sensor accuracy contribute to the oscillatory behavior. Nevertheless, as a whole, the system responds stably to the depth step command.

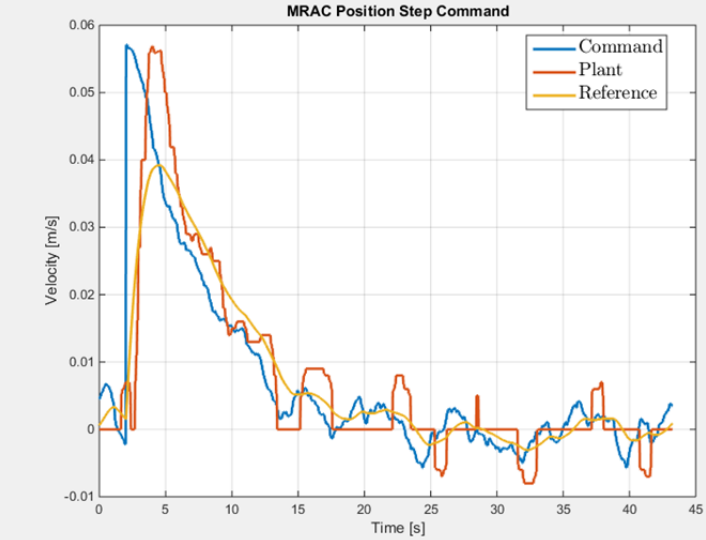

Figure 35. MRAC: Depth Step Command

Although the error value in Figure 36. does not converge, it does remain conservatively within the reasonable limits of measurement noise. This assertion is supported by the similarly small magnitude of command signals. For comparison, average velocity error is 0.0106 m/s, INS velocity accuracy is 0.001 m/s and DVL velocity accuracy is 0.2 m/s. Based on this analysis, the error is practically assumed to converge to zero.

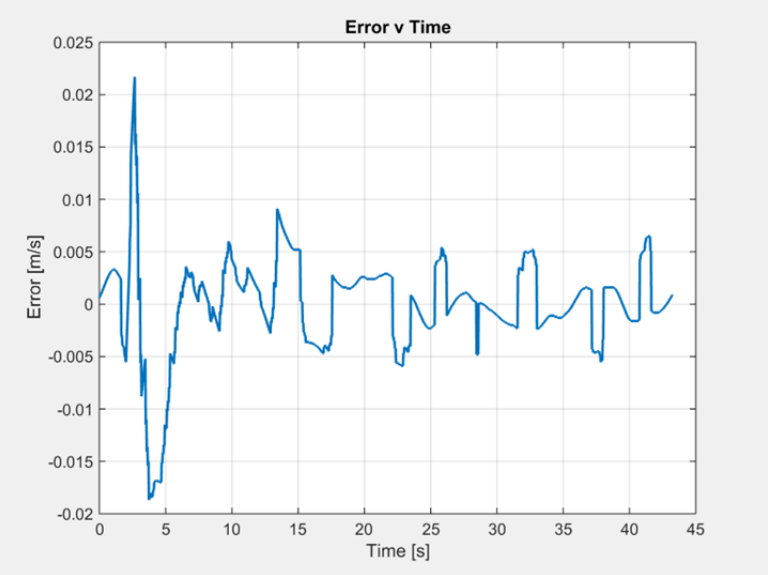

Figure 36. MRAC: Depth Step Command Error

The step command has no effect on the controller parameters. The parameters were allowed to reach steady state solutions prior to the step command and, as is evident in Figure 37. remain constant during the system response. This behavior correlates to a constant system configuration.

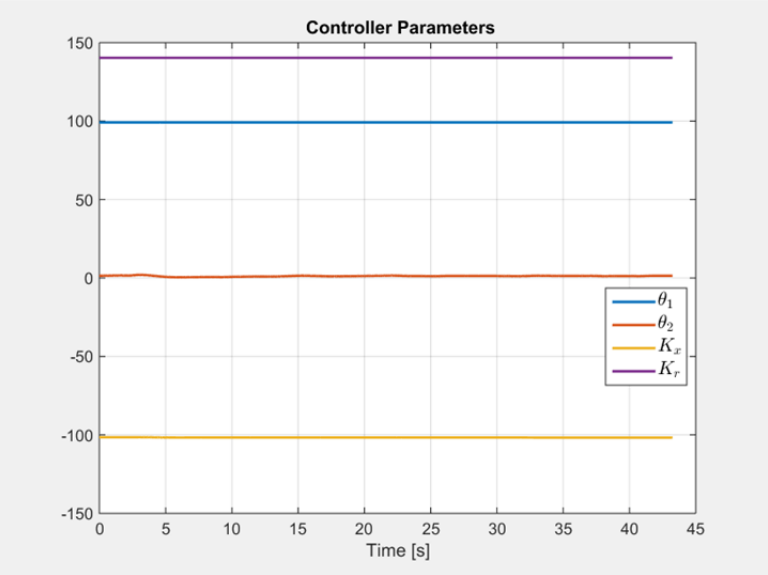

Figure 37. MRAC: Depth Command Error Controller Parameters

The control signal (Figure 38.) passed to the THAUS remains well bounded and achievable. Undesirable oscillations persist as a product of position error, sensor accuracy, and noise.

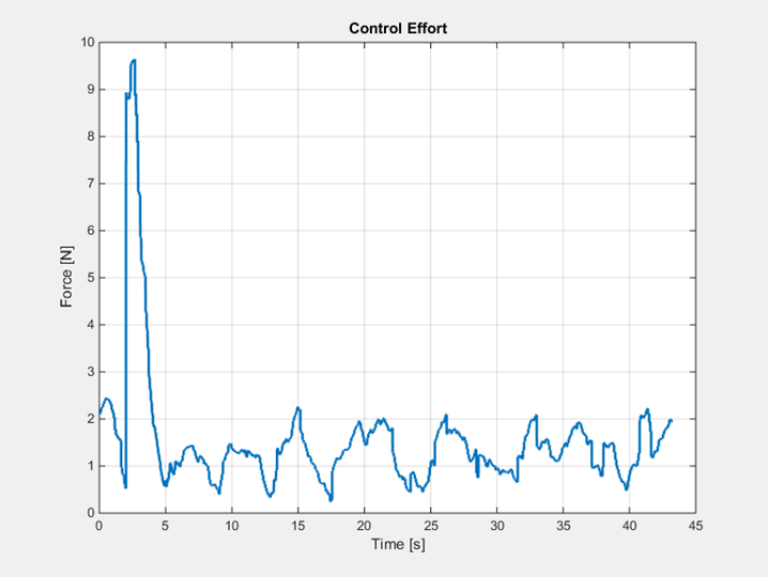

Figure 38. MRAC: Depth Step Command Control Signal

3. MRAC Weight Added

In order to evaluate the MRAC performance for a configuration change, a 0.5 kg weight was added to the vehicle while depth command was $z = 0.1$ m. All other parameters remained the same as in the previous experiment. Here, in Figure 39. , the system response to the plant configuration change is evaluated. Velocity initially increases as the extra weight begins to sink the vehicle. Adaptive corrections are then quickly made and the vehicle compensates for the extra weights, returns to its ordered depth, and velocity converges to zero.

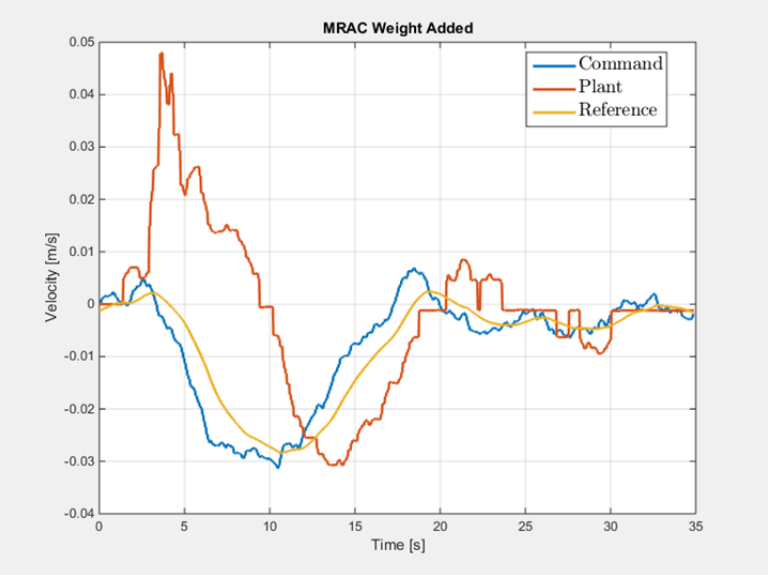

Figure 39. MRAC: Weight Added System Response

The system error in Figure 40. changes quickly when the weight is added and the vehicle sinks, recovers and increases in the opposite direction, but then settles near zero as expected.

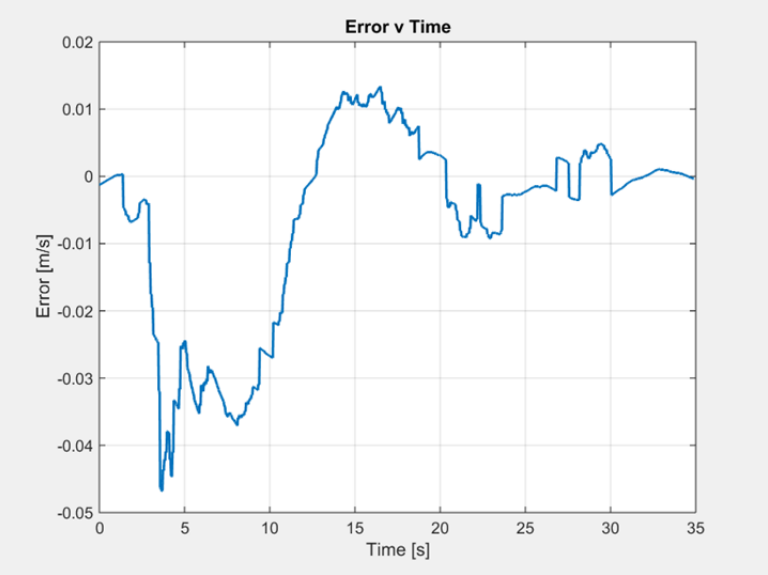

Figure 40. MRAC: Weight Added Error

The majority of the system's controller parameters, shown in Figure 41. , are unchanged since plant dynamics relating to dampening are unaffected by the added weight. The θ_2 parameter, which corresponds to the restoring forces, $g_z(\eta)$, does change though, to compensate for the deviation from neutral buoyancy.

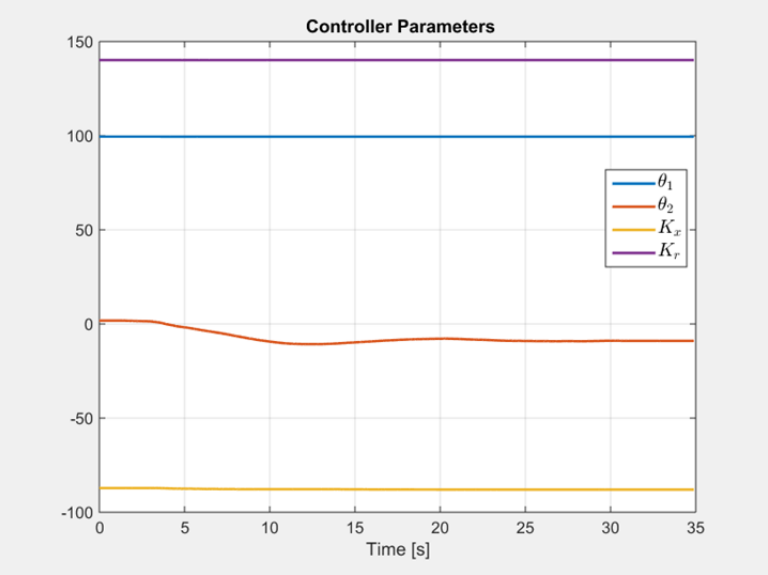

Figure 41. MRAC: Weight Added Controller Parameters

Control effort remains bounded as the system adapts and the control signal converges to a new steady state value. Inspection of the data in Figure 42. shows that the vehicle changed from a slightly positive buoyance to negative buoyancy.

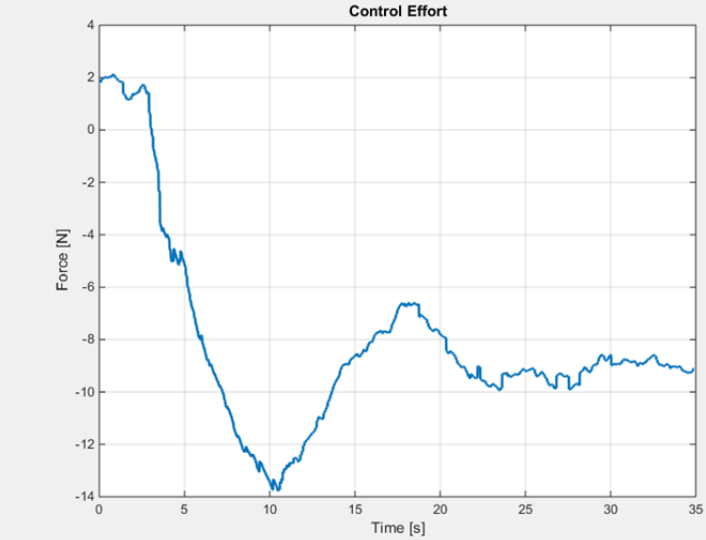

Figure 42. MRAC: Weight Added Control Effort

4. MRAC Weight Removed

The configuration change experiment was repeated in reverse, this time removing the 0.5 kg weight. The vehicle and controller parameters remained constant in all other regards. All of the results observed exhibited the expected behavior. The vehicle accelerated toward the surface after the weight was removed but quickly slowed its assent and returned to its ordered position. System response is shown in Figure 43.

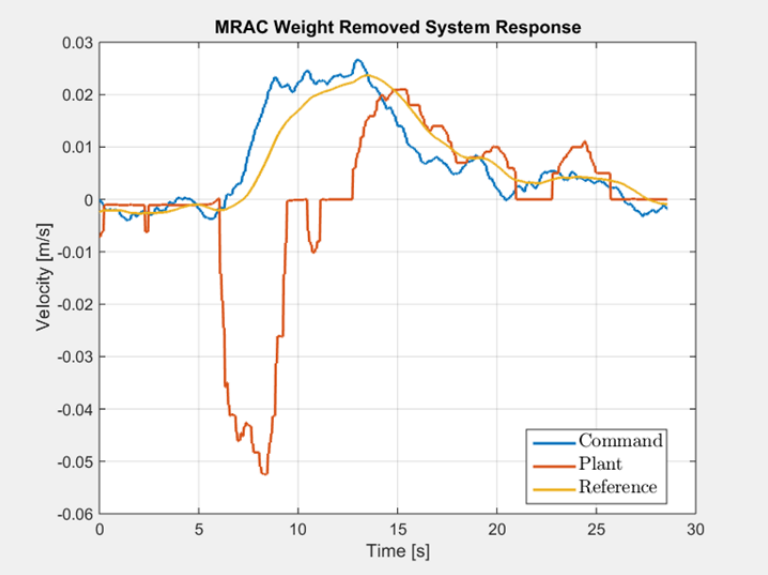

Figure 43. MRAC: Weight Removed System Response

The error (Figure 44.) correspondingly increase immediately after the weight is removed but then quickly begins converging back to zero as parameter adaptation takes place.

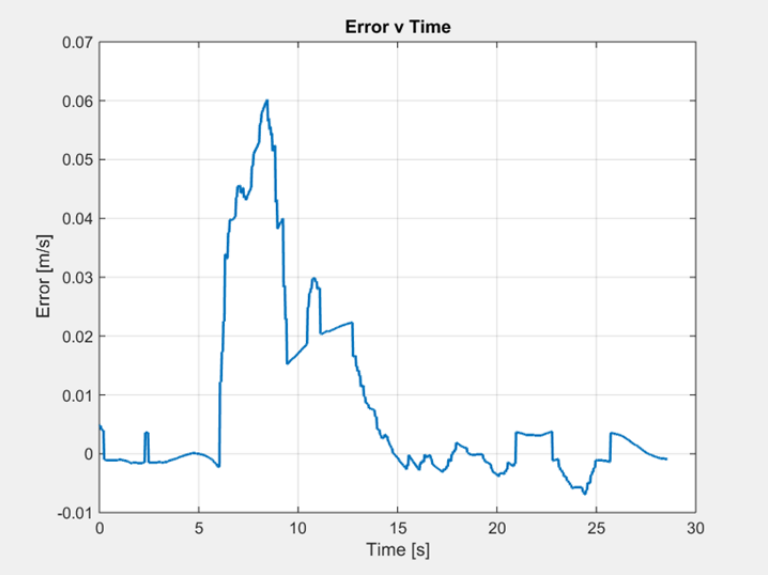

Figure 44. MRAC: Weight Removed Error

In Figure 45. parameter adaptation of θ_2 can be clearly seen. After the weight is removed the value trends back to the original steady state value near zero, corresponding to the near neutral buoyancy condition.

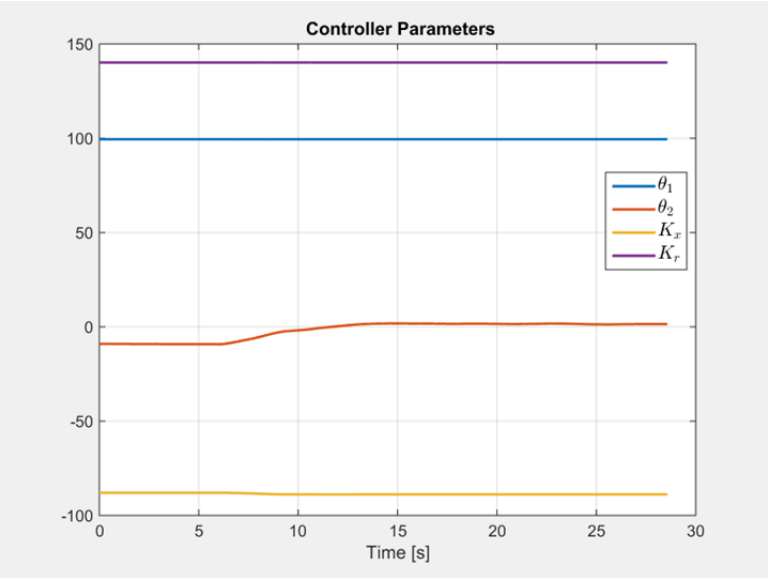

Figure 45. MRAC: Weight Removed Controller Parameters

The control effort, in Figure 46. , after the weight is removed settles at the previous, steady state level for the vehicle configuration with slightly positive buoyancy.

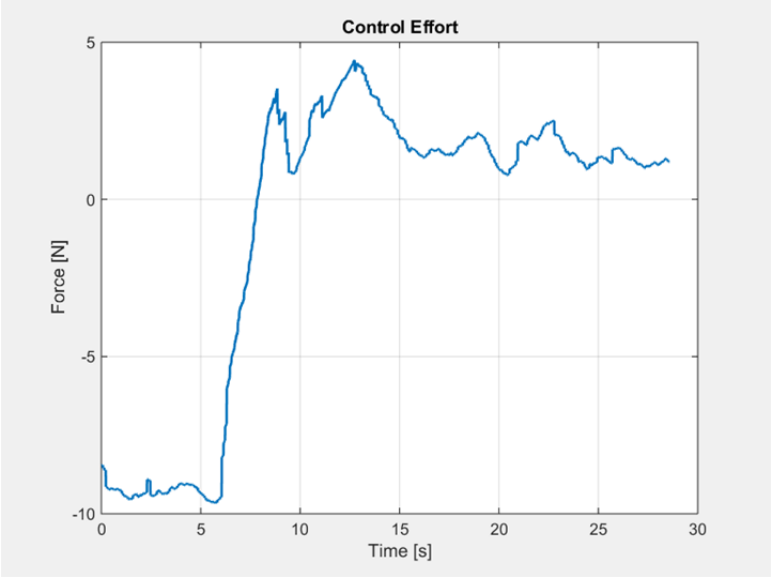

Figure 46. MRAC: Weight Removed Control Effort

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY

The proposed Robotic Diver's Assistant System would augment a dive team in their underwater operations to enhance safety, efficiency and capabilities of joint robot-human missions in arduous environments. A RDAS requires a robust, quickly adaptable control system to enable a THAUS to operate accurately and stably in dynamic environments, in close quarters with human divers, and in a variety of configurations that may change mid-mission.

The proposed platform for the RDAS is the vLBV300 THAUS. Because this AUV is operated within 4-DOFs the vehicle can be precisely controlled and is a prudent candidate for the RDAS. Building on previous research the THAUS was modeled using experimentally derived velocity step response data. The parametric dynamic model derived for this system was written in terms of a regressor. The Linear Least Squares SysID method was applied to the system in order to identify a nominal parametric dynamic model for use in development of an adaptive control system. The two candidate control systems were Model Reference Adaptive Control and $\mathcal{L}_1$ Adaptive Control.

The MRAC system proved to be very capable in simulation. The adaptive architecture enabled the system to adapt the controller parameters in the presence of unmodeled, inaccurately modeled, and time variant parameters. Simulation experiments demonstrated that the adaptive system enabled the THAUS to consistently track a constantly changing velocity as well as maintain a constant velocity, both with small steady state errors. The system was capable of converging to stable controller parameter solutions without accurate initial conditions of those parameters. This resulted in successful tracking that was achieved without a priori knowledge of the plant's dynamic parameters or the controller parameters. Tracking performance proved to be very accurate over time; however, coupling prevented the system from both performing robustly and adapting quickly. Adaptation gains had to be selected such that robustness was not excessively diminished in order to achieve fast adaptation.

This limitation motivated the addition of modifications to increase the robustness in the system while still enabling fast adaptation. The dead zone modification disabled the adaptation process when the state error was sufficiently low. This prevented the adaptation law unnecessarily searching for controller parameter solutions once a “good enough” solution had been found. The dead zone modification reduced oscillation and drift in controller parameters, enabling more stable system tracking. The Projection Operator modification bounded the controller parameters within selected limits preventing so called parameter “wind-up,” a more severe form of parameter drift. The Projection Operator would redirect the solution vector for a controller parameter if the algorithm determined that parameter lay outside of the projection boundaries or within the projection tolerance. The Projection Operator modification prevented controller parameter solutions with excessively high values and the resulting system becoming unstable. Enforcing parameter bounds reduced the occurrence of oscillatory behavior both in frequency and amplitude. Additionally, the control efforts required to track the reference system were reduced. In combination these two modifiers allowed the system to accept high adaptation gains required for fast adaptation, while maintaining robustness, producing good steady state performance in both transients and steady state.

In preparation for implementing the MRAC system onto the THAUS, the effects of data transfer rate transitions on system performance was investigated. Asynchronous transition functions were introduced into the MRAC Simulink model to simulate the different data rates provided by the controller and the vehicle sensors. The rate transitions limited the control system’s ability to quickly respond to incremental changes, because slower, asynchronous data transfer prevented the system from recognizing those changes had occurred. Then, the system could not quickly send the updated control signal based on those changes. As a result the robustness and steady state tracking ability of the system suffered, necessitating a reduction in adaptation gains

Implementation of the MRAC system onto the vLVB300 THAUS for physical experiments was accomplished via introduction of a ROS based interface. The controller was first evaluated in response to step commands with successful performance. The system proved capable of accurately tracking the reference model despite the lack of a

priori knowledge of plant dynamics. The controller was also evaluated in response to plant parameter variation by changing the mass of the vehicle. The control system performed successfully in this task as well. The change in the plant configuration was quickly recognized and the adaptation law adjusted the controller parameters as required to compensate for this change, enabling the system to maintain accurate steady state tracking. Overall, the MRAC system proved to be both capable and practical; however, the limitations of the system with regard to adaptation speed and robustness persisted. The transient response of the system was particularly less stable than desired. This limitation became especially apparent in physical implementation, even with the use of the modifiers. An MRAC system is a great candidate for an RDAS but must be meticulously tuned to meet performance expectations.

In light of the limitations in MRAC, an alternative adaptive control solution was evaluated: $\mathcal{L}_1$AC. This control system decouples robust performance from fast adaptation. $\mathcal{L}_1$AC accomplishes decoupling by implementing a low pass filter on the control signal input to the state predictor and plant. This filter eliminates high frequency content associated with selection of high adaptation gains. High frequency content in the control signal can contribute to a loss of robustness evidenced by oscillations, overshoot and overly aggressive corrections in the system response. In addition to the filter, robustness is aided by the use of the Projection Operator in the adaptation law. As in MRAC, the Projection Operator bounds the controller parameters according to selected limits. These key features of $\mathcal{L}_1$AC enable it to perform exceptionally well with regard to both robustness and adaptation speed. Indeed, for a properly formulated problem stability and parameter convergence can be guaranteed.

The simulated system responded extremely well to the sinusoidal velocity command in simulation. The plant began tracking with the state predictor almost immediately. Error remained very low through the simulation, and control effort was limited to within achievable bounds. Additionally, controller parameters converged around steady state values. A similarly well performing response was observed in the simulation with time-varying parameters and a step input command. System response to

the step command was fast and smooth, with very little error between the state predictor and the plant, both of which performed similar to the ideal plant model representing the desired response. Adaptation parameters converged quickly and responded very quickly to the change in plant parameters caused by increasing the system mass. Again, the control effort was well bounded to within achievable values.

The results were not without imperfections, though. High frequency oscillations of small amplitude were observed in the control signal. This high frequency content did remain stable, however, and did not cause oscillations or unstable behavior in the plant and state predictor response. This behavior emphasizes the importance of selecting an appropriate control signal filter. The filter selected must be meticulously selected to ensure that high frequency content is eliminated and the system response is not affected. Overall the system performed extremely well is should be considered as a candidate for further implementation on the physical platform.

B. FUTURE WORK

The development of the RDAS using the vLBV300 is an ongoing research effort by CAVR. Further research in the field of adaptive control systems will continue to benefit the project and will support the vehicle's envisioned mission.

More advanced modeling of the vLBV300 is in progress. The advanced methods being used to model the vehicle will create a more accurate, coupled dynamic model. This model will aid in simulation, control system design, and evaluation.

With regard to the development of the adaptive controllers, continued careful tuning of control and adaptation parameters is required to ensure the controller is operated optimally. MRAC could also benefit from research into additional modifications that could enhance robustness. $\mathcal{L}_1$AC requires more careful attention to the control signal filter in order to eliminate high frequency content.

Continued experimental evaluation of the MRAC is recommended. Additionally, $\mathcal{L}_1$AC should be implemented onboard the THAUS as MRAC was. Evaluating these two control architectures in parallel will ensure the selection of the control system candidate

best suited for the mission of a RDAS. In this vein, more quantitative (vice qualitative) analysis techniques should be employed to support selection of control system architecture and associated parameters.

To further support the choice of the controller and its parameters, more experimental research should be conducted in the environment the system is anticipated to operate in (i.e., ocean conditions with currents, waves, etc.). Exposure to external disturbances and other unknowns are an excellent way to prove the capabilities of the control systems. Finally, the techniques and procedures exhaustively perfected in regard to the heave channel will be replicated for the other degrees of freedom, producing a more comprehensive, multi-input multi-output adaptive control system that can robustly stabilize the coupled dynamical system.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. MRAC CODE AND SIMULATION DIAGRAM

A. MATLAB CODE

```
%% MRAC with Time-Varying Parameters
% LT Nicholas Valladarez
% Naval Postgraduate School
% December 2014

clear all; close all; %clc;
%% System Parameters

% Dynamic Properties
m1 = 33; m2 = m1 + .02; % Mass [kg]
buoy = m1*9.81;         % Buoyancy [N]
mg = [m1*9.81 m2*9.81]; % Weight [N]
Z_dw = -244.3629;
Z_ww = -128.8485;
a1 = [0 0];
a2 = [Z_ww/(m1-Z_dw) Z_ww/(m2-Z_dw)];
a3 = [1/(m1-Z_dw) 1/(m2-Z_dw)];

% Reformating Model
A = [a1(1)]; B = [a3(1)]; C = [1];
Am = [-0.4];
kg = -1/(C'/Am*B);
Bm = B*kg;

% Controller Parameters
P = lyap(Am',eye(1));
PB=P*B;
delta = 0.001;          % Dead zone
Epsilon = 0.3;          % Projection Tolerance
Gamma_x = 1e5;          Pr_x = 100;     Kx_0 = 0;     % Gain / Proj
Bound / IC
Gamma_r = 1e5;          Pr_r = 200;     Kr_0 = 0;     % Gain / Proj
Bound / IC
Gamma_theta = 1e5;      Pr_t = 50;      Theta_0 = 0;  % Gain / Proj
Bound / IC

%% Simulation

% Simulation variables
xm0 = 0;            % Model initial condition
x0 = 0;             % Plant initial condition
r = .5;             % Signal Amplitude
TIME = 100;         % Simulation Time
TRIP = 50;          % Parameter variation time
SAT = 85;           % Control signal saturation value
NOISE = 0;          % Noise Power
FREQ = .1;          % Sin wave frequency
```

```
SWITCH = 2;          % Signals: 1 = Sin Wave; 2 = Step; 3 = Ramp; 4 =
Pulse;

% Run Simulation
sim('MRAC_sim_TVP')
```

B. SIMULINK DIAGRAM

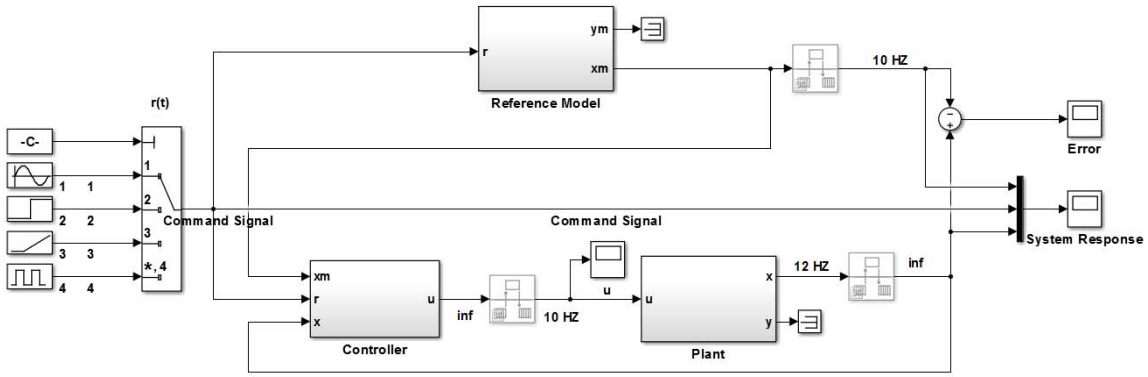

Figure 47. MRAC: Simulink Model

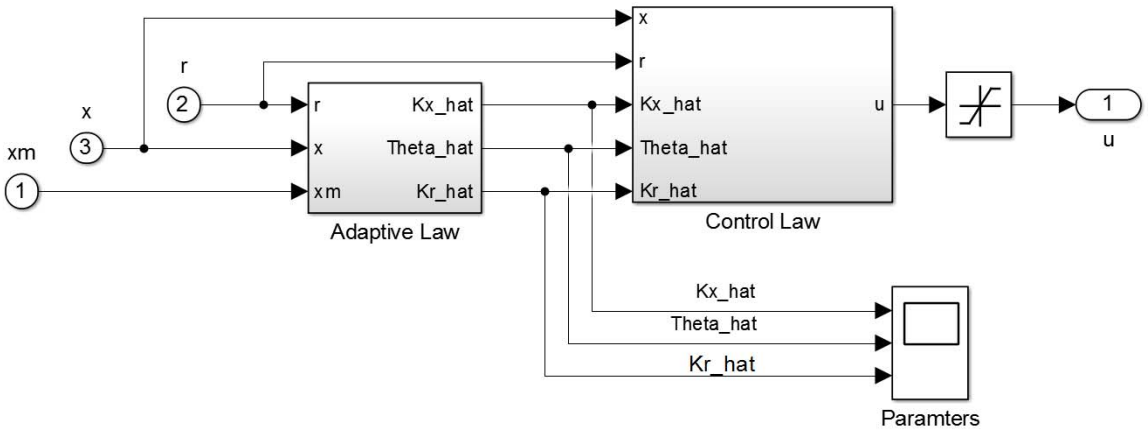

Figure 48. MRAC: Controller

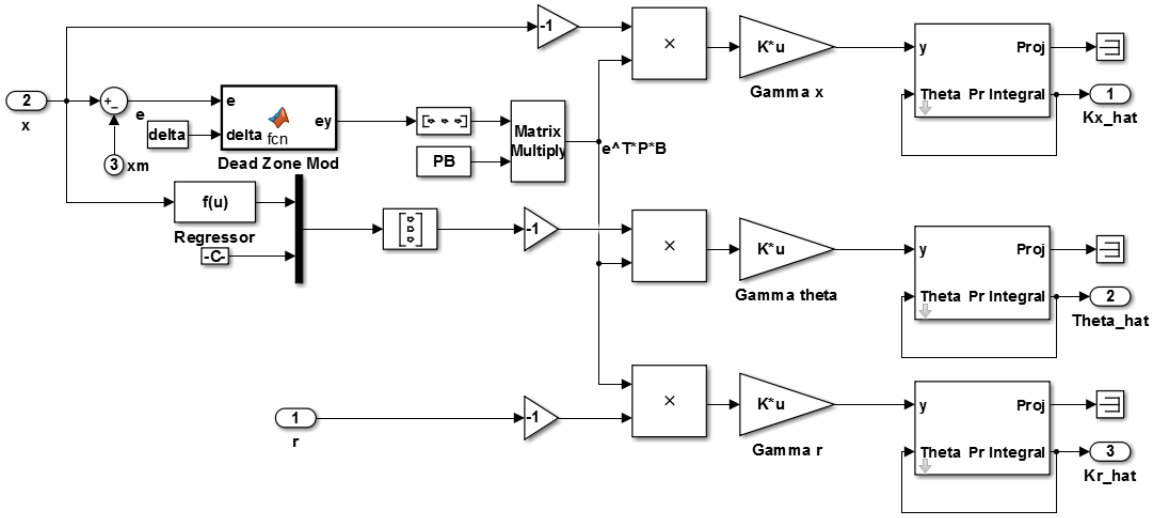

Figure 49. MRAC: Adaptation Law

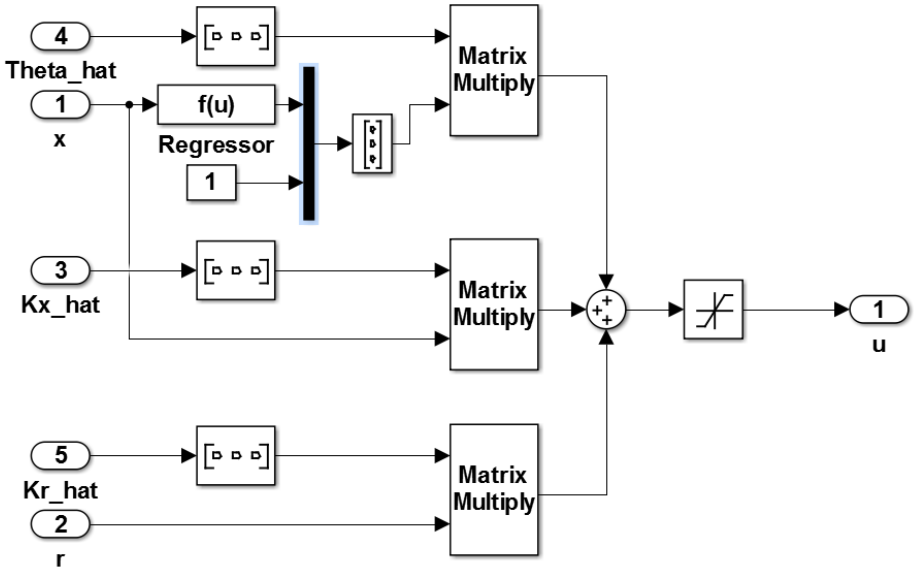

Figure 50. MRAC: Control Law

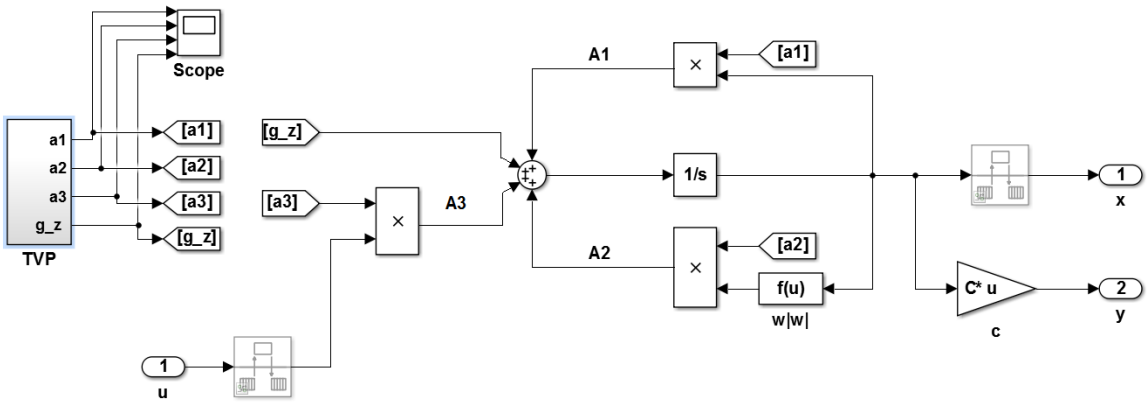

Figure 51. MRAC: Plant

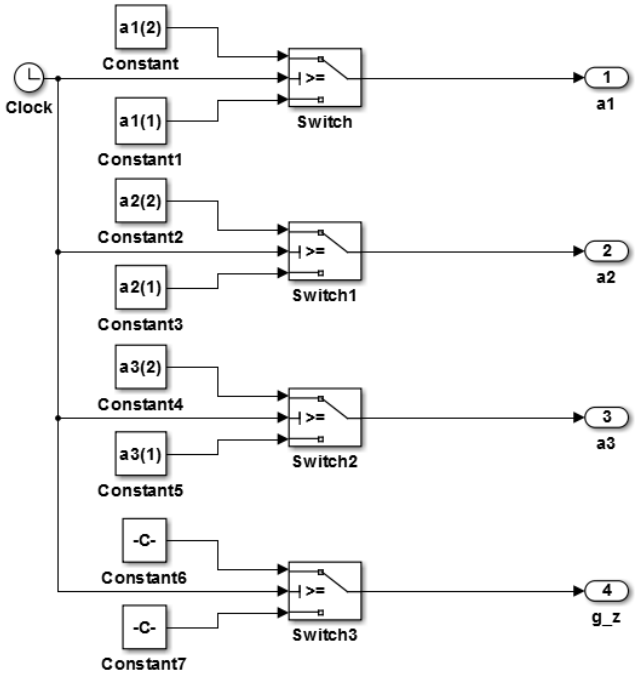

Figure 52. MRAC: Time-Varying Parameters Switch

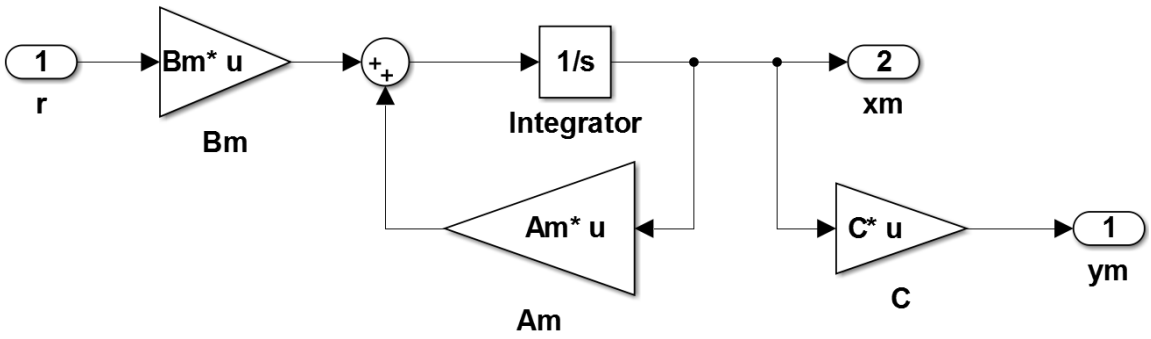

Figure 53. MRAC: Reference Model

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. $\mathcal{L}_1$ AC CODE AND SIMULATION MODELS

A. MATLAB CODE

```
%% L1 AC with Time-Varying Parameters
% LT Nicholas Valladarez
% Naval Postgraduate School
% December 2014

clear all; close all; clc;
%% System Parameters

% Dynamic Coefficients
m1 = 33.0; m2 = m1 + .02;
buoy = m1*9.81;              % Buoyancy [N]
mg = [m1*9.81 m2*9.81];      % Weight [N]
Z_dw = -244.3629;
Z_ww = -128.8485;
a1 = [0 0];
a2 = [Z_ww/(m1-Z_dw) Z_ww/(m2-Z_dw)];
a3 = [1/(m1-Z_dw) 1/(m2-Z_dw)];

% Reformating Model
A = [a1(1)]; B = [a3(1)]; C = [1];
Am = [-0.4];
kg = -1/(C'/Am*B);
Bm = B*kg;

% Controller Parameters
P=lyap(Am',eye(1));
PB=P*B;
Gamma = 5e7;
k=6;
Ds_n=[1];   % Filter
Ds_d=[1 0]; % Filter
K_x_id = (Am-A)/B;  % Ideal gain
%K_r_id = Bm/B;      % Ideal gain

% Parameter Bounds
Epsilon = 0.3;                      % Projection Tolerance
Omega=[0.3 2.0];                    % Bounds on actuator efficiency
Omega_m = (Omega(2)+Omega(1))/2;    % Omega proj center
Omega_r = (Omega(2)-Omega(1))/2;    % Omega proj bounds
Pr_s = 400;                         % Sigma proj bounds
Pr_t = 200;                         % Theta proj bounds
K_s0 = 0;                           % Sigma proj IC
K_t0 = 0;                           % Theta proj IC
K_o0 = 0;                           % Omega proj IC


%% Simulation
```

```
% Simulation Variables
xhat0 = 0;          % State estimate IC
x0 = 0;             % State IC
r=0.5;              % Command amplitude
TIME = 30.0;        % Simulation time
TRIP = 15;          % TVP time
SAT = 85;           % Control signal saturation value
NOISE = 0;          % Noise Power
FREQ = .1;          % Sin wave frequency
SWITCH =2;          % Signals: 1 = Sin Wave; 2 = Step; 3 = Ramp; 4 =
Pulse;

% Run Simulation
sim('Heave_L1_TVP_sim');
```

B. SIMULINK DIAGRAM

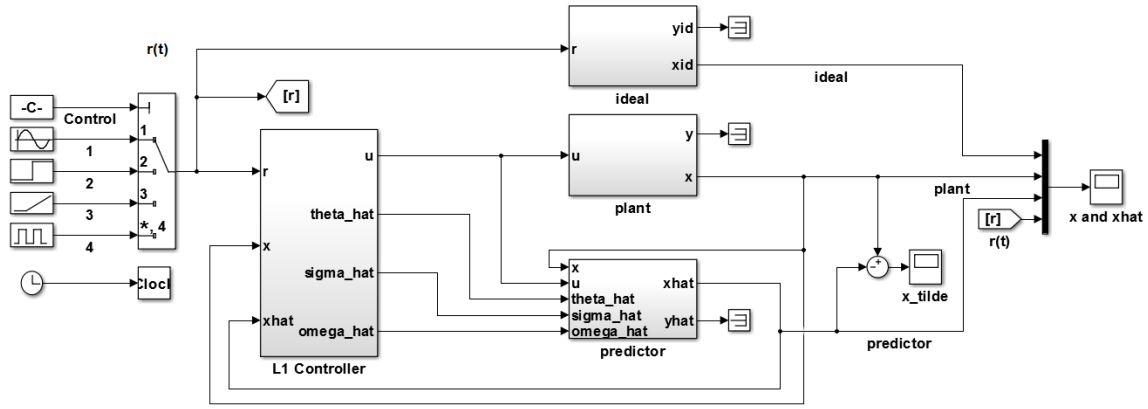

Figure 54. $\mathcal{L}_1$ AC Simulink Model

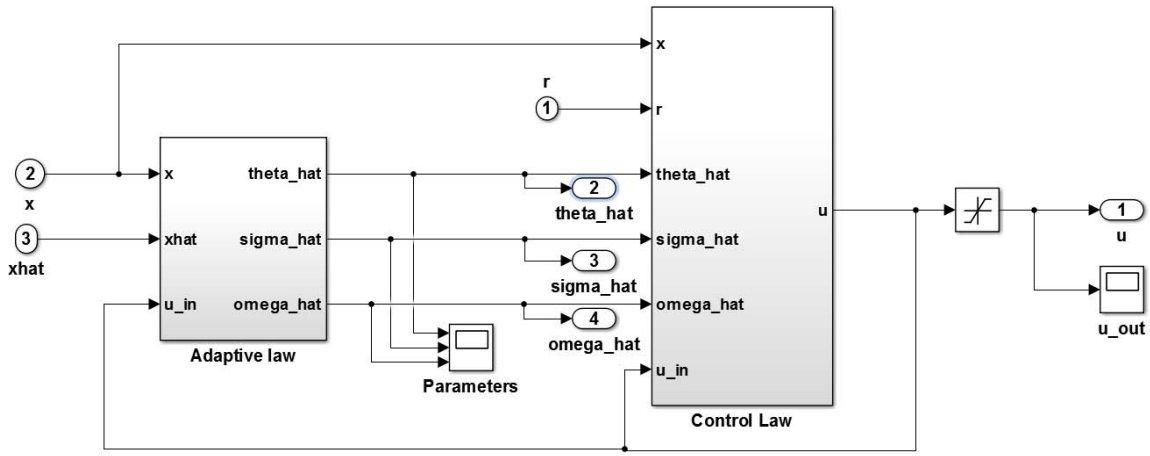

Figure 55. $\mathcal{L}_1$ AC: Controller Model

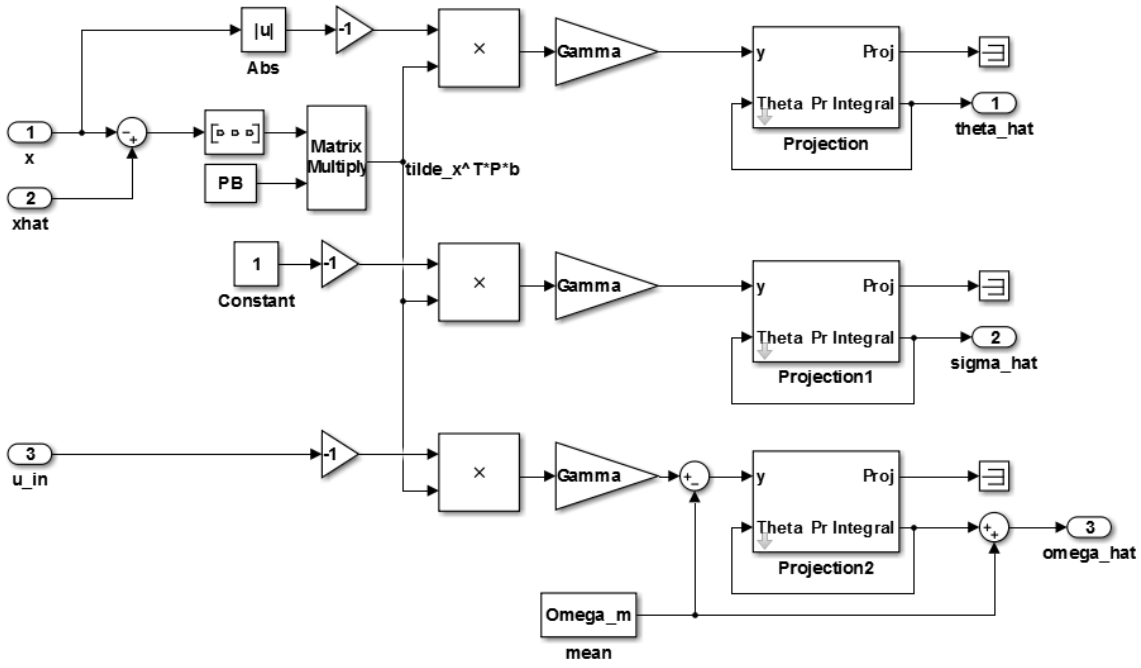

Figure 56. $\mathcal{L}_1$ AC: Adaptation Law

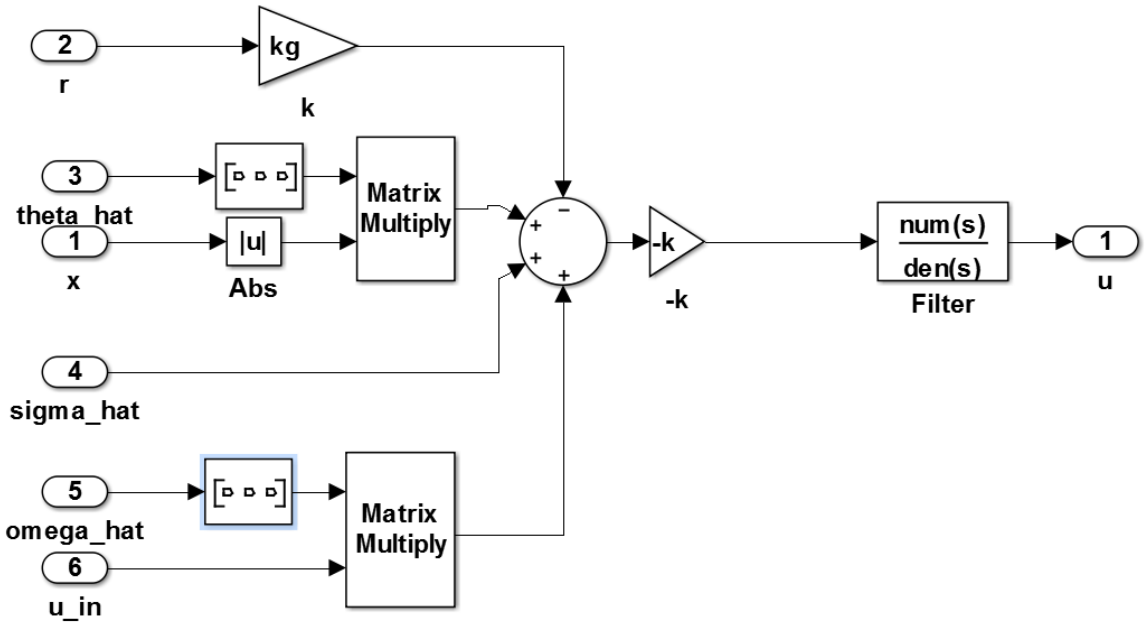

Figure 57. $\mathcal{L}_1$ AC: Control Law

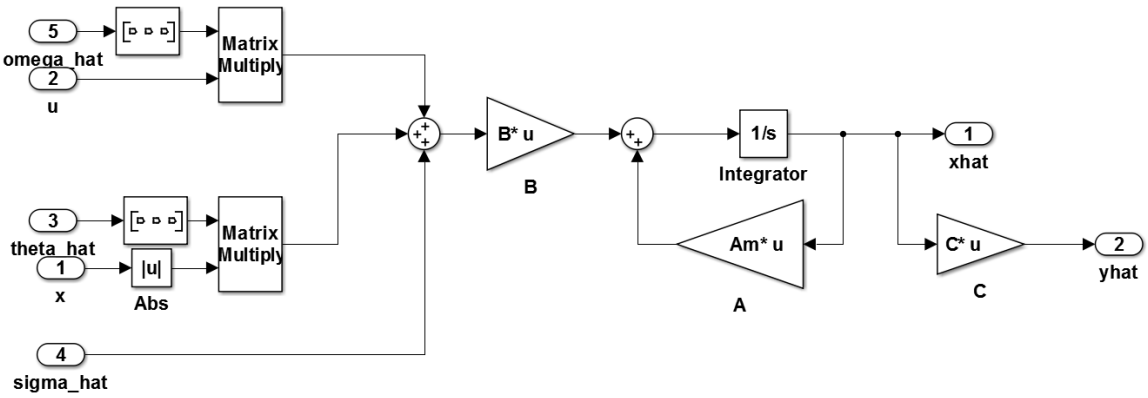

Figure 58. $\mathcal{L}_1$ AC: State Predictor

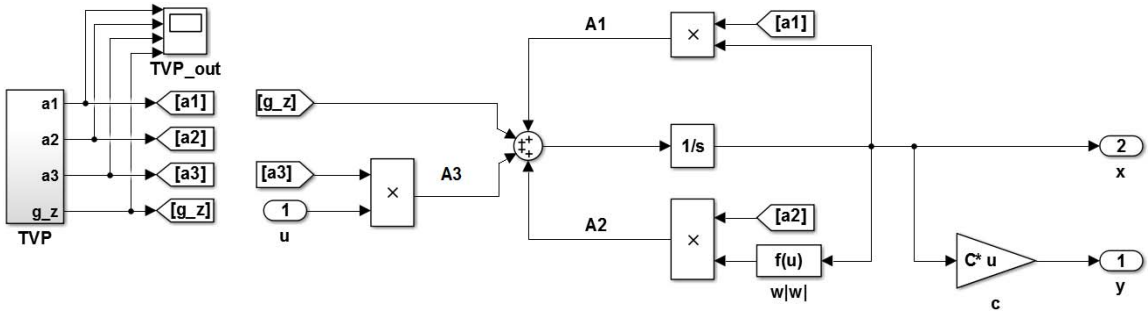

Figure 59. $\mathcal{L}_1$ AC: Plant

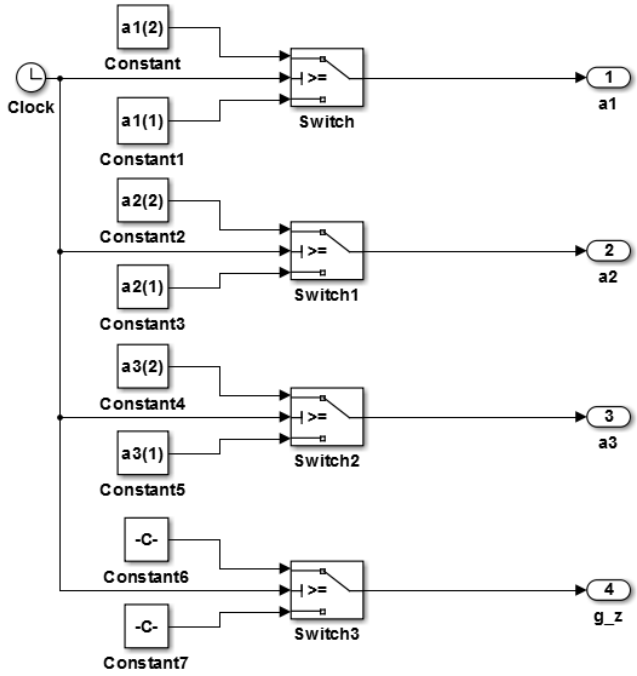

Figure 60. $\mathcal{L}_1$ AC: Time-Varying Parameters Switch

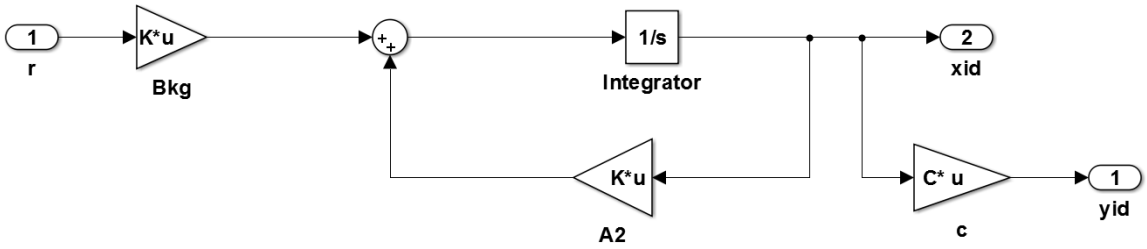

Figure 61. $\mathcal{L}_1$ AC: Ideal Plant

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. DERIVATION OF k_g

For a system of form:

$$\dot{x} = A_m x + B(u_{ad} + f(x,t))$$
$$y = Cx$$

assume $u_{ad}(s) \triangleq C(s)(k_g R(s) - \hat{F}(s))$ and $C(0) = 1$, where $\hat{F}(s) \triangleq \mathscr{L}\{\hat{f}(x,t)\}$, $F(s) \triangleq \mathscr{L}\{f(x,t)\}$, and $C(s)\hat{F}(s)$ cancels the effects of $F(s)$. The steady state error is

$$e(t) \triangleq r(t) - y(t)$$
$$= r(t) - Cx(t)$$

and

$$E(s) \triangleq \mathscr{L}\{e(t)\} = R(s) - Y(s) = R(s) - CX(s)$$

The Laplace transform of $\dot{x}$:

$$sX(s) = A_m X(s) + B(U_{ad}(s) + F(s))$$
$$= A_m X(s) + BC(s)k_g R(s) - \cancel{BC(s)\hat{F}(s)} + \cancel{BF(s)}$$

Solving for $X(s)$:

$$(s - A_m)X(s) = BC(s)k_g R(s)$$

$$X(s) = (s - A_m)^{-1} BC(s)k_g R(s)$$

Substituting $X(s)$ into $E(s)$:

$$E(s) = R(s) - CX(s)$$
$$= (1 - C(s - A_m)^{-1} BC(s)k_g)R(s)$$

According to the final value theorem:

$$\mathscr{L}\{\lim_{t\to\infty} e(t)\} = \lim_{s\to 0} sE(s)$$

Also, to achieve zero steady state error for a step input:

$$R(s) = \frac{r_m}{s}$$

where r_m is the size of the step function.

$$\therefore \qquad \lim_{s\to 0} sE(s) = \lim_{s\to 0} \cancel{s}\,(1 - C(s - A_m)^{-1} B \cancel{C(s)^{1}} k_g)\frac{r_m}{\cancel{s}}$$

Therefore, for zero steady state error:

$$\boxed{k_g = \frac{-1}{CA_m^{-1}B}}$$

LIST OF REFERENCES

[1] K. J. Astrom and B. Wittenmark, *Adaptive Control.* Reading, MA: Addison-Wesley, 1995.

[2] T. Fossen, *Guidance and Control of Ocean Vehicles.* New York: John Wiley & Sons, 1994.

[3] J. Yuh, "Modeling and control of underwater robotic vehicles," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 20, pp. 1475–1483, November / December 1990, 1990.

[4] J. D. Weiss, "Real-time dynamic model learning and adaptation for underwater vehicles," M.S. thesis, Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, 2013.

[5] M. Caccia, G. Indiveri, and G. Veruggio, "Modeling and identification of open-frame variable configuration Unmanned Underwater Vehicles," *IEEE Journal of Oceanic Engineering,* vol. 25, pp. 227–240, 2000.

[6] A. T. Streenan, "Diver relative UUV navigation for joint human-robot operations," M.S. thesis, Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, 2013.

[7] J. E. Slotine and L. Weiping, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice Hall, 1991.

[8] R. Yang, B. Clement, A. Mansour, H. J. Li, M. Li, and N. L. Wu, "Modeling of a complex-shaped underwater vehicle," in *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Espinho, 2014, pp. 36–41.

[9] N. S. Nise, *Control Systems Engineering.* Hoboken, NJ: John Wiley and Sons, 2004.

[10] P. Ioannou and J. Sun, *Robust Adaptive Control.* Upper Saddle River, NJ: Prentice Hall, 1996.

[11] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control with Aerospace Applications.* London: Springer-Verlag, 2013.

[12] J. E. Slotine and Weiping Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research,* vol. 6, pp. 49–59, September 1, 1987.

[13] S. Nicosia and P. Tomei, “Model reference adaptive control algorithms for industrial robots,” *Automatica,* vol. 20, pp. 635–644, 9, 1984.

[14] T. Fossen and S. Sagatun, “Adaptive control of nonlinear underwater robotic systems,” in *1991 IEEE International Conference on Robotics and Automation,* Sacramento, CA, 1991, pp. 1687–1694.

[15] D. Maalouf, I. Tamanaja, E. Campos, A. Chemori, V. Creuze, J. Torres, and L. Rogelio, “From PD to nonlinear adaptive depth-control of a tethered autonomous underwater vehicle,” in *IFAC Joint Conference 2013, 5th Symposium on System Structure and Control,* Grenoble, France, 2013, .

[16] C. C. Cheah, C. Liu, and J. J. E. Slotine, “Adaptive Tracking control for robots with unknown kinematic and dynamic properties,” *The International Journal of Robotics Research,* vol. 25, pp. 283–296, March 01, 2006.

[17] A. Sanei and M. French, “A non-singular performance comparison between two robust adaptive control designs,” in *40th IEEE Conference on Decision and Control,* Orlando, Florida, 2001, pp. 2043.

[18] C. Wen, J. Zhou, Z. Liu, and H. Su, “Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance,” *IEEE Transaction on Automatic Control,* vol. 56, pp. 1672–1678, 2011.

[19] Chengyu Cao and N. Hovakimyan, “Design and analysis of a novel adaptive control architecture with guaranteed transient performance,” *IEEE Transactions on Automatic Control,* vol. 53, pp. 586–591, 2008.

[20] D. Maalouf, V. Creuze, and A. Chemori, “A novel application of multivariable $\mathcal{L}1$ adaptive control: From design to real-time implementation on an underwater vehicle,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference On,* Vilamoura, 2012, pp. 76–81.

[21] *vLBV300 MiniROV Systems*. (2014, October 20). SeaBotix Inc. [Online]. Available: http://www.seabotix.com/products/vlbv300.htm.

[22] *BlueView P-450 Series*. (2014, October 20). BlueView Technologies, Inc. [Online]. Available: http://www.blueview.com/products/2d-imaging-sonar/p450-series/.

[23] *BOLTON - Fully Integrated Navigation and Control*. (2014, October 20). Greensea Systems, Inc. [Online]. Available: http://greenseainc.com/greensea_bolton_datasheet_rev2_0.pdf.

[24] *Explorer Doppler Velocity Log*. (2014, October 20). Teledyne Technologies, Inc. [Online]. Available: http://www.rdinstruments.com/explorer.aspx.

[25] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.

[26] E. Kharisov, N. Hovakimyan, and K. J. Astrom, "Comparison of several adaptive controllers according to their robustness metrics," in *AIAA Guidance, Navigation and Control Conference*, Toronto, Ontario, Canada, 2010.

[27] C. Cao and N. Hovakimyan, "L1 adaptive controller for nonlinear systems in the presence of unmodelled dynamics: Part II," in *2008 American Control Conference*, Seattle, WA, 2008, pp. 4099.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California